MARZO 86 - 300 ptas.

Todospectrum

AÑO II – Número 19

REVISTA EXCLUSIVA PARA USUARIOS

INVESTIGAMOS

Los hackers al descubierto

OIMOS

El generador de sonido

PROBAMOS

Los nuevos comandos del 128

Suplemento QL

PROTECCION DEL SOFTWARE

La REVISTA de las computadoras personales que interesa tanto al aficionado como al profesional

ORDENADOR POPULAR

La informática al alcance de todos.

COMPLETA, DETALLADA, DIRECTA, INNOVADORA...

Una publicación de clara y amena lectura.

Para el aficionado, para el profesional. Técnica, diversión y entretenimiento.

ORDENADOR POPULAR

la revista que **le dá a conocer el hoy y el mañana de la informática.** ¡Con las últimas novedades en el campo de las computadoras personales!



¡NO SE QUEDE ATRAS!

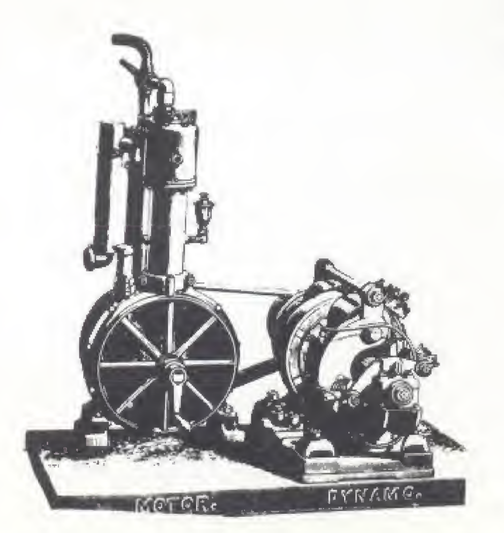
YA ESTA A LA VENTA

Cómprela en su kiosco habitual o solicítela a:

ORDENADOR POPULAR

Bravo Murillo, 377
Tel. 733 96 62
28020 MADRID

AÑO II - Nº 19 - MARZO 1986





SERVICIO DE EJEMPLARES ATRASADOS

Complete su colección de

Todospectrum

A continuación le resumimos el contenido de los ejemplares aparecidos hasta ahora.

Núm. 2 - 300 ptas.
Gráficos profesionales/Desplazamiento pixel a pixel/Utilización de rutinas/Construcción del interface centronics/Programas de utilidad para microdrive/Rutina reset en código máquina/Análisis del editor de textos Tasword/Interfaces para impresoras/Programas.

Núm. 3 - 300 ptas.
Novedades sonimag'84/Ampliando el Basic/Programas para ordenar programas/Gráficos con el VU-3D/Lenguaje Forth/Archivos en microdrive/Programación de un interface de impresora/Programas.

Núm. 4 - 300 ptas.
De profesión: programador/Consola para el Spectrum/Comparación código máquina-Basic/Análisis programa contabilidad/Calendario/Pascal/Programas.

Núm. 5 - 300 ptas.
Floppys para Spectrum/Diseño asistido por ordenador/64 Caracteres por linea/Juego de la vida/Pascal/Asi hacemos las portadas/Control de evaluaciones/Programas.

Núm. 6 - 300 ptas.
Representación de funciones/Todos los caminos conducen a la ROM/Juegos/Pascal/Construcción de un lápiz óptico/Programas de gestión. El SITI/Logo: torgugas para todos/ Interrupciones del Z-80/Programas.

Núm. 7 - 300 ptas.
Del 48 al PLUS paso a paso/¿Plotter para Spectrum?/Juegos/Libros de código máquina/Lápiz óptico. Programación del montaje/El LOGO en la escuela/Pascal/Floppys para Spectrum/Programas.

Núm 8 - 300 ptas.
Amplía tu memoria... a 48 K/Arquitectura: análisis del PREYME/Juegos/FORTH. Nociones básicas/Una clave, please/QL Magazine. Ultimas novedades, análisis de software, Lenguajes/Aula informática con Spectrum/Programas.

Núm 9 - 300 ptas.
Spectrum parlanchín/Juegos/Aula informática con Spectrum/Análisis: Comercial 4/Pascal/Periféricos: Wafadrive/QL Magazine: EASEL lo mejor de PSION. Música con QL/Desplazamiento Pixel a Pixel, aportación de lectores/Programas/Programer II.

Núm. 10 - 300 ptas.
Discos: invesdisc 200/Juegos/Dos programas simultáneos/Protección del software/Conozca extremadura, consulte a su ordenador/Desensamblador Z-80/Sofware educativo/QL Magazine: novedades Informat, Hoja de cálculo, Ajedrez/Construya su propio Joystick/Pascal/programas.

DISPONEMOS DE TAPAS ESPECIALES PARA SUS EJEMPLARES DE ZX
(sin necesidad de encuademación)

Núm. 11 - 300 ptas.
Actualidad/La otra cara del LOGO/Juegos/El Spectrum habla castellano/SOFTAID ayuda para Etiopía/S.O.S. aquí el Spectrum/Dibujar con lápiz óptico/QL Magazine: Procesador de textos. Teclas de función programables/Programas.

Núm. 12 - 300 ptas.
Actualidad/Inteligencia artificial/Lápiz óptico dk'TRONICS/Juegos/Análisis/Bingo/Z-80 PIO/Código máquina/Análisis: MASTERFILE/Programas.

Núm. 13 - 300 ptas.
Actualidad/Discos: Discovery 1/Juegos/Inteligencia artificial/Un nuevo sistema operativo/QL Magazine: Archive, Cartridge doctor. Aplicaciones comerciales/Código máquina/Programas.

Núm. 14 - 300 ptas.
Actualidad, Spectrum 128/Cálculo de estructuras para ingenieros y arquitectos/HELP utilidades en microdrive/Juegos/El microdrive ese desconocido/Código máquina/QL Magazine: GRAPHIC QL. Juegos. Discos de 720 K/Un nuevo operativo/Programas.

Núm. 15 - 300 ptas.
Actualidad/Spectrum 128/Un nuevo operativo/Circulos redondos/Juegos/Utilidades: BETA-BASIC/QL Magazine: Introducción al SUPER BASIC. Nuevas utilidades/Hardware: Puertas lógicas/Código máquina/Programas.

Núm. 16 - 300 ptas.
Actualidad/Cinco horas con SCREEN$/Hardware práctico/Cálculos de infinita precisión/Juegos/Un nuevo operativo/QL Magazine: Gráficos en SUPER-BASIC. Dibujando con ratón. Archivos con Archive. Programa/La última batalla, Juego estratégico.

Núm. 17 - 300 ptas.
Actualidad/Gráficos interactivos/Juegos/Código máquina/Un nuevo operativo/Trucos de programación/QL Magazine: Radiografía del QL. Gráficos en SUPER-BASIC/Libros/Programmas.

Núm. 18 - 300 ptas.
Actualidad/Introducción al C/Libros/Juegos/De cinta a microcinta/Visión panorámica de los microprocesadores más comunes/QL Magazine: Copy de grises. Microprocesadores 68000, una familia numerosa/Curioseando en la ROM/Programas.



Para hacer su pedido, rellene este cupón HOY MISMO y envíelo a:

Todospectrum Bravo Murillo, 377
Tel. 733 96 62 - 28020 MADRID

Ruego me envíen los siguientes ejemplares atrasados de TODOSPECTRUM .... al precio de 300 pts

El importe lo abonaré
☐ POR CHEQUE ☐ CONTRA REEMBOLSO ☐ CON MI TARJETA DE CREDITO ☐ AMERICAN EXPRESS ☐ VISA ☐ INTERBANK

Numero de mi tarjeta ....

Fecha de caducidad .... Firma

NOMBRE ....

DIRECCION ....

CIUDAD .... C. P. ....

PROVINCIA ....

DIRECTOR:
Enrique F. Larreta
REDACTOR JEFE:
Emiliano Juárez
REDACCION:
Ignacio Borrell, Octavio López,
Antonio del Río
DISEÑO:
Ricardo Segura y Benito Gil

Editado por PUBLINFORMATICA, S. A.
Presidente:
Fernando Bolin
Director Editorial Revistas de Usuarios:
Juan Arencibia
Gerente de circulación y ventas:
Luis Carrero
Producción: Miguel Onieva
Director de Marketing:
Antonio González
Servicio al cliente:
Julia González. Tel.: 733 79 69
Administración:
PUBLINFORMATICA, S. A.
Publicidad:
Emilio García
Dirección y Redacción:
Bravo Murillo, 377. 5.º A. Tel. 733 74 13.
Télex: 48877 OPZX e 28020 Madrid
Administración y Publicidad:
Bravo Murillo, 377. 3.º E. Tels.
733 96 62 - 96
Publicidad Barcelona:
María del Carmen Ríos. Pelayo, 12. Tel.
(93) 318 02 89. 08001 Barcelona
Depósito legal: M-29041-1984
Distribuye S.G.E.L. Avda. Valdelaparra,
s/n. Alcobendas (Madrid).
Fotomecánica: Karmat, C/ Pantoja, 10.
Madrid.
Fotocomposición: Artecomp.
Imprime: Héroes, C/ Torrelara, 8. Madrid.
Distribuidor en VENEZUELA,
SIPAM, S. A.
AVD. REPUBLICA DOMINICANA, EDIF.
FELTREC - OFICINA 4B BOLEITA SUR
CARACAS (VENEZUELA)
Esta publicación es miembro de la
Asociación de Revistas de
Información **ARI** asociada a la
Federación Internacional de Prensa
Periódica, FIPP.

SUSCRIPCIONES:
Rogamos dirijan toda la correspondencia
relacionada con suscripciones a:
TODOSPECTRUM EDISA: Tel. 415 97 12
C/ López de Hoyos, 141-5º
28002 MADRID
(Para todos los pagos reseñar solamentes
TODOSPECTRUM)
Para la compra de ejemplares atrasados
diríjanse a la propia editorial
TODOSPECTRUM
C/ Bravo Murillo, 377. 5.º A
Tel. 733 74 13 - 28020 MADRID

Si deseas colaborar en TODOSPECTRUM
remite tus artículos o programas a Bravo
Murillo, 377. 5.º A. 28020 Madrid. Los
programas deberán estar grabados en
cassette y los artículos mecanografiados.
A efectos de remuneración, se analiza cada
colaboración aisladamente, estudiando su
complejidad y calidad.

EDITORIAL

NOS LLEGO EL CAMBIO



Como anunciábamos el mes pasado, este número de *Todospectrum* ofrece numerosas novedades. Indudablemente el lector ya habrá reparado en una de ellas: la portada. Desde sus comienzos, hace año y medio, esta revista se distinguió por ser la única cuya cubierta se realizaba con un ordenador: el Spectrum, como no podía ser menos. Pues bien, cumplido sobradamente el propósito de mostrar las posibilidades gráficas de la máquina, creemos que ha llegado el momento de presentar una portada más elaborada y atractiva, que aporte más información sobre la revista.

Además, teniendo en cuenta la lógica evolución de los usuarios de ordenadores Sinclair, hemos introducido cambios que afectan tanto al diseño como al contenido de los artículos.

El plato fuerte de este número es un extenso trabajo sobre los hackers.

Guiados por Manuel Arana, colaborador habitual de *Todospectrum,* nos adentramos en una de las actividades características de todo hacker que se precie: la modificación de programas.

Y no faltan las secciones habituales de código máquina, lectores, actualidad, juegos y programas, ni artículos, como los dedicados a los nuevos comandos del 128 o al acceso al BASIC desde código máquina.

Completa la revista el suplemento QL Magazine, en el que se analizan dos temas de actualidad: la protección del software y los entornos operativos controlados por iconos, como el ICE.

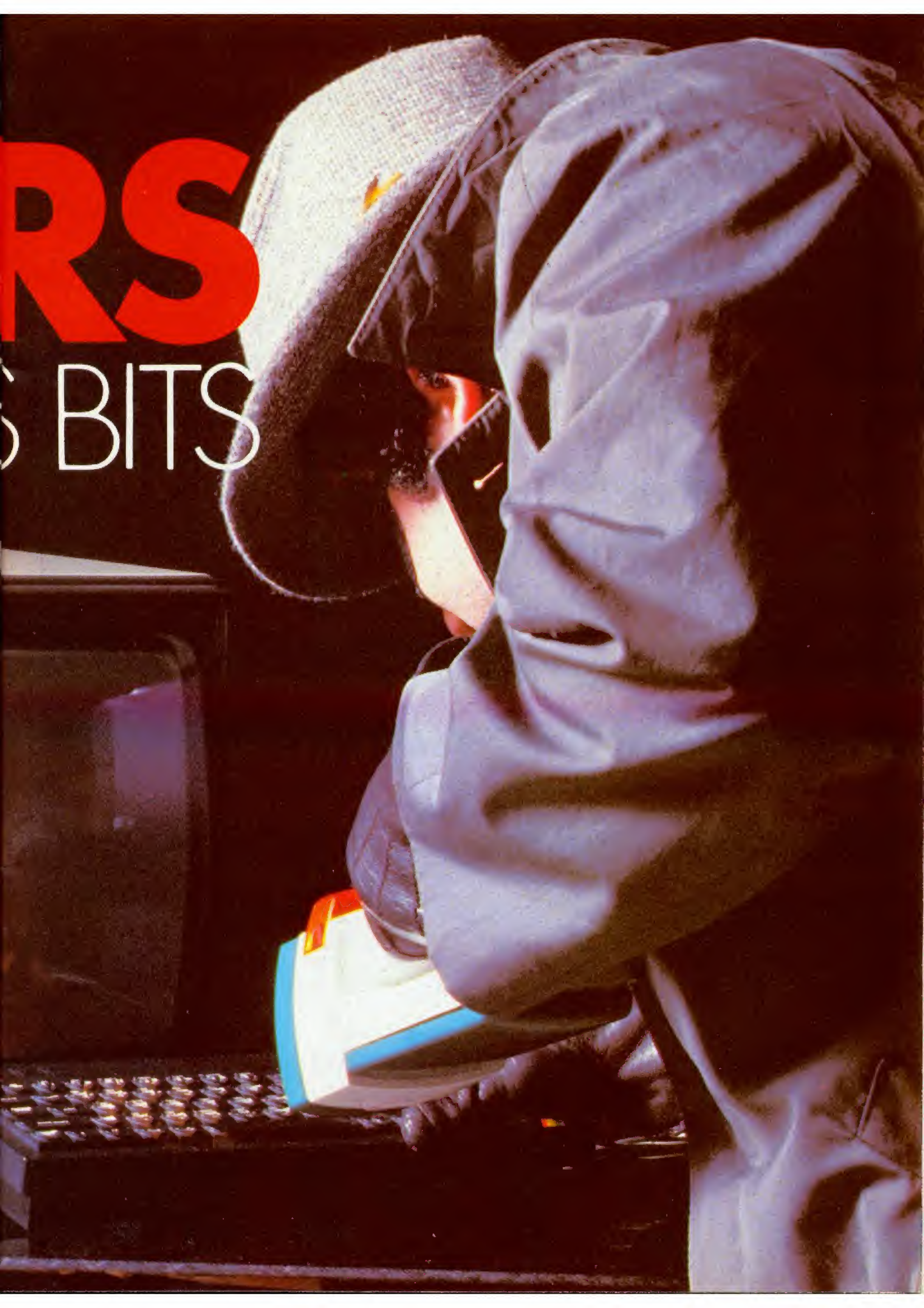
Invitamos a los lectores a que nos hagan saber su opinión sobre los cambios realizados y los que introduciremos próximamente.

TODOSPECTRUM

HACKE
LOCOS POR LOS

Seguramente, los lectores de esta revista han visto la película «Juegos de Guerra» u otros del mismo tema. Esos chicos algo pícaros y algo genios forman parte de una generación nacida con la proliferación de los micrordenadores. Se llaman asimismo hackers y se empecinan en jugar a comunicarse con otros ordenadores, generalmente muy grandes.




RS
BITS

El término *hacker* ha empezado a ser conocido hace poco debido a la aparición de la película «Juegos de Guerra» y a la detención por el FBI de algunos chicos que habían logrado meterse en los ordenadores del Pentágono. Pero sus orígenes son muy antiguos, incluso anteriores a la aparición de ordenadores personales.

Los auténticos orígenes de los *hackers* se pueden encontrar en unos dispositivos denominados «cajas azules». La aparición de éstos se debió, entre otras cosas, a un error de concepto de la compañía telefónica americana, **Bell** (denominada hoy **ATT**). Sobre los años cincuenta esta empresa realizó la planificación de cómo iba a ser el funcionamiento futuro de toda la red telefónica estadounidense y eligió como patrón un sistema basado en tonos para enviar toda la información por la línea telefónica.

Lo peor del caso es que en un alarde de ingenio, publicaron todos los datos (tipo de información, duración y frecuencia de tonos, etc.) en un boletín de difusión pública. Cuando se dieron cuenta de su error e intentaron retirarlo, era demasiado tarde.

El primero en descubrir la utilidad de esto fue un chico ciego llamado **Joe Engressia.** Para no aburrirse cuando se quedaba solo en casa, solía llamar a aquellos números que tenían instalado un contestador automático y silbaba canciones que quedaban grabadas. Pero una vez, haciéndolo, el contestador se puso automáticamente en marcha y le dejó escuchar todos los mensajes grabados anteriormente. Una rápida llamada a la compañía telefónica despejó sus dudas. Había silbado justo el tono que le indicaba al contestador que reprodujese los mensajes. Evidentemente esto le incitó a seguir probando y comentó los resultados a todos sus amigos.

La película «Juegos de Guerra» logró impactar en los espectadores dando a conocer la realidad de los «hackers»

Poco a poco, ya en la década de los sesenta y principios de los setenta, empezaron a surgir grupos de personas que en un afán de diversión o de ahorrarse dinero diseñaron dispositivos que emitían estos tonos y que por tanto les permitían jugar con la red telefónica. Estos aparatos se conocían popularmente como cajas azules y las personas que los diseñaban y usaban jugarían un papel muy importante en la aparición de los ordenadores personales y del concepto *hacker,* aunque el nombre que recibían era ligeramente distinto, *phreakers.*

Entre toda esta gente había dos chicos que se acababan de conocer y que se habían hecho amigos debido a la gran propensión que tenían ambos por las bromas. Ambos se llamaban Steven y como el lector habrá adivinado, sus apellidos eran (y son) Jobs y Wozniak. Posteriormente fundarían una compañía de ordenadores conocida como **Apple Computer.** Con los conocimientos de uno (Wozniak) y la charlatanería de otro (Jobs) lograron obtener cierta fama y hacer un buen número de amigos entre los *phreakers* de la época.

Uno de los amigos que hicieron se llamaba **John Draper** y era ampliamente conocido por sus habilidades telefónicas hasta que fue detenido por el FBI y condenado. Posteriormente escribiría un procesador de textos para el **Apple II** y en vista de su relativo éxito sería contratado por otra compañía, pese a su oscuro pasado, para hacer el mismo programa para un nuevo ordenador personal. La compañía se llama **IBM** y el programa es el famoso **Easy Writer.**

La fiebre de las cajas azules siguió hasta que la **Bell** («Ma Bell» como suelen decir los estadounidenses) tomó acciones judiciales. En ese momento y de resultas de estas presiones, la fiebre *phreaker* empezó a declinar. Declive que hubiera ocultado toda la historia bajo una capa de polvo si no fuera porque a un investigador medio



loco llamado **Ed Roberts** le dio por inventar un nuevo aparato llamado **Altair 8800**, que fue el primer ordenador personal propiamente dicho.

Los primeros ordenadores

Rápidamente empezó otra fiebre que arrastró consigo a la mayoría de los expertos en cajas azules a este nuevo campo todavía inexplorado. Estos primeros ordenadores distaban mucho de lo que es el ordenador personal de hoy en día. No tenían discos y la mayoría de las veces tampoco salida de video. Sólo unos interruptores y luces que permitían programarlos en lenguaje máquina. Por ello, mercado principal no fueron los negocios, sino los *hobbistas.* En muchos casos eran construidos por los propios usuarios y los planos se distribuían de un modo totalmente informal en reuniones y charlas.

La gente que construía y jugaba con estos aparatos empezó a ser conocida como *hackers,* término no traducible pero que viene a significar «tipo que está grillado por los ordenadores». En general un *hacker* era una persona que se divertía jugando con un ordenador del mismo modo que los naturalistas se van al campo a ver animales. Muchos de los programas y empresas famosas hoy en día surgieron de la unión de *hackers* para un proyecto común.

Un ejemplo típico lo constituye un chico llamado **Bil Gates;** en unión de varios amigos hizo un intérprete de BASIC, lo que constituyó un avance importante. Este intérprete se empezó a distribuir ampliamente y pronto fue lenguaje estándar. En vista del éxito **Bill Gates** fundó una compañía que se ha convertido en la importante casa de *software* **Microsoft.**

La paciencia y tesón de este gente sirvió para hacer avanzar la tecnología hasta conseguir



modelos más avanzados. Casas más o menos grandes empezaron a interesarse en este mercado (**Tandyu, Atari,** etc.), y las pequeñas empezaron a crecer, como **Apple.** Todo esto condujo a la aparición de ordenadores más potentes.

A su vez los *hackers* unieron sus nuevos conocimientos a los adquiridos previamente con las líneas telefónicas y empezaron a comunicarse usando un aparato denominado *modem* que convierte las señales eléctricas del ordenador en sonidos que pueden ser enviados por el teléfono y viceversa. Esto era potenciado por la liberal legislación norteamericana en materia de comunicaciones que permitía conectar cualquier aparato a la red si cumplía los requisitos básicos para no dañar la línea. Rápidamente empezaron a surgir clubs y los denominados BBS *(Bulletin Board Systems)* que no eran más que ordenadores con *modems* especiales que podían responder automáticamente al teléfono.

Al principio los BBS eran establecidos por los *hackers* en sus propios ordenadores haciéndolos funcionar por la noche (hora ideal en la que suceden la mayor parte de las aventuras informáticas) y el uso de la red telefónica se limitaba a comunicarse con los BBS, pero pronto las cosas empezaron a cambiar.

Exploradores telefónicos

En Estados Unidos y debido a la estructura de las compañías, es bastante común que las empresas con grandes ordenadores tengan estos conectados a la red telefónica normal para que los directivos y otro personal puedan trabajar con ellos desde sus casas con un terminal y un *modem*. Esto, naturalmente, también posibilita que cualquier otra persona que disponga de los mismos materiales (o incluso mejor, un ordenador en vez de una terminal) pueda llamar al mismo número y conectarse. Evidentemente los ordenadores están protegidos contra estos asaltantes y piden unas claves de acceso que son secretas, pero ya se sabe que todo lo que una persona ha protegido, otra lo puede desproteger, de modo que rápidamente mucha gente empezó a intentar meterse en estos ordenadores sin permiso. Este tipo de accesos, protagonizados por una pequeña minoría del colectivo *hackers*,



UNA CONVERSACION FICTICIA

Uno de los principales objetivos de estos «bucaneros» son los grandes ordenadores, jugar con ellos es una de las aficiones más extendidas de esta comunidad y para ello se pasan horaras intentando traspasar las barreras que les impiden la entrada.

Imaginemos que son las 12 de la noche en el centro de cálculo de un gran banco. La mayoría de los programadores hace horas que se han ido a sus casas y sólo están las tres personas encargadas de velar por el funcionamiento del sistema durante la noche. La mayoría de los días su trabajo se limita a comprobar que todo marcha correctamente y que los programas que tienen menos prioridad se ejecuten ahora que hay menos trabajo.

Hasta ahora el trabajo se ha desarrollado con normalidad y una charla amigable se desarrolla con el ruido de fondo de las unidades de disco. De pronto en la consola del sistema (normalmente una impresora con un teclado) aparece un mensaje. Esto normalmente se hace cuando algún usuario tiene un problema y pide ayuda al encargado, pero a estas horas de la noche es muy extraño, por lo que los tres se acercan a mirar. El texto es escueto:

!HOLA????

(El símbolo / es uno de los denominados *promt* del sistema y en nuestra conversación indica algo que se imprime, asimismo el símbolo > indica que el encargado del sistema teclea lo que viene a continuación.)

La sorpresa es bastante considerable, este tipo de mensajes no es normal y menos a estas horas de la noche. Después de algunos segundos de pausa aparece otro.

>HAY ALGUIEN AHI?

Por fin el encargado del turno se decide a contestar y teclea

> MSG "QUIEN ES?"

Con MSG indica que quiere enviar un mensaje (lo entrecomillado) a la otra persona.

/ SOY UN USUARIO. Y TU?

> MSG "SOY EL ENCARGADO DEL SISTEMA. POR FAVOR INDIQUE SU CUENTA Y QUE HACE A ESTAS HORAS"

Al cabo de algunos segundos en los que el terminal se queda silencioso, aparece la respuesta.

son los que han dado fama al nombre y han hecho que actualmente se conozca como *hacker* al individuo que se introduce ilegalmente en otros ordenadores. Una consecuencia inmediata ha sido que la palabra haya perdido parte de su significado inicial para el público, aunque dentro del auténtico grupo de entusiastas se sigue sabiendo que un *hacker* es un loco por los ordenadores que es capaz de pasarse la noche sin dormir (con o sin *modem*) con tal de acabar un programa porque le apetece y no tiene por qué hacer nada prohibido, o de un modo más general, aquéllos que trabajan con ordenadores porque les gusta y no por ganar dinero. De todos modos el resto de este artículo lo dedicaremos a aquéllos que han alcanzado la fama gracias a sus exploraciones prohibidas y que para diferenciar de los anteriores llamaremos «bucaneros».

A la vista de lo interesante que podía ser un gran ordenador y con el sabor de lo prohibido en la boca, muchos *hobbistas* se lanzaron a hacer exploraciones nocturnas, sin salir de casa, en pos de trofeos, los ordenadores de las grandes compañías.

Steven Jobs, Wozniak, Ed Roberts, Bill Gates y John Draper son míticos nombres dentro de la leyenda de los hackers

Durante largas horas estos personajes tecleaban en sus ordenadores diversas combinaciones que les permitieran acceder a estos sistemas, ¡y a veces lo conseguían! Una vez dentro se dedicaban a investigar cómo era el ordenador, qué sistema operativo tenía, etcétera. También buscaban cosas interesantes, que en la mayoría de los casos no era la información confidencial de la empresa, sino programas curiosos. También utilizaban estos ordenadores para crear con ellos BBS para el grupo de amigos. En el apartado «Una conversación ficticia» podemos ver un ejemplo típico de lo que sucede cuando un *hacker* entra en un ordenador después de derribar sus barreras.

«Juegos de Guerra»

El impacto se produjo hace dos años con la aparición de la película «Juegos de Guerra», en la que el protagonista primero estaba a punto de provocar una guerra nuclear desde su casa (involuntariamente por supuesto) y luego lo evitaba sólo con sus conocimientos sobre ordenadores.

En estos últimos dos años ha sucedido gran cantidad de cosas que han traído el tema a la actualidad.

En primer lugar la película, seguida de numerosos arrestos por el FBI. Un ejemplo son «El hombre eco» (16 años), «Tres

/ SOY JOHN DEL DEPARTAMENTO DE CONTABILIDAD, CUENTA $CONT1, ESTAMOS HACIENDO BALANCE PROVISIONAL.

Este hecho no es demasiado raro y ya se ha producido otras veces, pero en esta ocasión no les han avisado de que se iban a quedar a trabajar, por lo que deciden realizar una investigación adicional. Uno de los operadores se sienta y empieza a teclear órdenes rápidamente. La primera le muestra todas las cuentas (usuarios y procesos del sistema) que están activas en este momento.

> PROCESOR STATUS
/ NUCLEUS 0001 00:00 234
/ MEMCON 0002 01:00 103
/ TERCON 0005 02:01 21
/ BATCON 0005 03:01 32
/ SYSMGR 0020 09:02 64
/ OPER1 0025 15:02 64
/ BATCH1 0090 45:03 128



/ BATCH2 0090 47:03 256
/ TEST 0030 75:02512
/ IUCONT 0070 61:03 35
/ CONT1 0070 82:02 64

Esto les permite comprobar rápidamente que efectivamente existe una cuenta CONT1 y que funciona de acuerdo con las prioridades asignadas, pero el otro operador se fija en algo muy curioso. La cuenta TEST se creó al instalar el sistema para realizar unas pruebas y aunque todavía existe, nadie la usa desde hace mucho. Además está consumiendo mucha memoria y tiene una prioridad alta, por lo que decide «espiar» lo que hace su usuario. Se sienta en su terminal y teclea.

> MONITOR $TEST
/
/ MONITOR $SYSMGR

Esto muestra que el usuario de TEST está espiándoles a ellos, actitud nada normal, por lo que rápidamente decide echarle del sistema sospechando, y no sin razón, que es una persona no autorizada.

> FORCE $TEST LOGOFF

Pero el ordenador contesta de un modo sorprendente.

/ PRIVILEGE VIOLATION

Indicando que no le está per- •••

piedras» (15 años) y «Tío Sam» (17 años), todos de California, que con sus ordenadores lograron meterse en el *mainframe* de la compañía telefónica y obtener llamadas gratuitas a cualquier parte del mundo.

Otro caso curioso lo protagonizaron cinco chicos que se lograron introducir en los ordenadores de la NASA, considerados hasta entonces inviolables.

Asimismo se han descubierto hechos curiosos. En Phoenix se descubrió a dos BBS, uno manejado por un chico de 14 años apodado «Underworld» y otro por un profesor de escuela, «Casa divertida», la fórmula para hacer un explosivo químico.

Aunque parece que la mayoría de estos «bucaneros» son jóvenes menores de 18 años, el retrato robot elaborado por el FBI indica que los más peligrosos tienen alrededor de 25 años, ambiciosos, con una posición de fuerza y frustrados por no disponer de una oportunidad para ejercer su talento.

Desde entonces se producen casos de éstos con una rapidez increíble, y el Senado norteamericano ha tenido que ponerse a trabajar para sacar una legislación acorde, ya que como es habitual los hechos han soprepasado a las leyes.

Perseguidos, acosados por la justicia, estos «locos por los ordenadores» disfrutan burlando los más sofisticados sistemas de vigilancia y protección.

Hay que decir que este tema no es particular de Estados Unidos; ya se han producido varios casos de chicos que se han metido en Prestel, la red de ordenadores de la compañía telefónica británica de uso público bajo suscripción. Muchos «bucaneros» se introducen en ella con códigos robados o descubiertos por el método de prueba y error. Ese es el caso de **Robert Shifreen** y **Stephen Gold** que fueron detenidos *in fraganti.* Sucesos similares han pasado en Francia e incluso en Alemania.

Lo más aterrorizante del tema (si se quiere decir así) es que la edad media de estos «bucaneros» no se coloca en los veintitantos años de un ingeniero graduado, sino que son jóvenes de 17, 15 o incluso 13 y 14 años los que protagonizan las mayores aventuras. No obstante se les suele considerar poco formados y se les intenta formar su personalidad. Casos notables de esto lo constituyen **Peter Leppik,** que después de ser arrestado y puesto en libertad condicional, ayudó a la policía a solucionar un problema con un programador (no *hacker*) que además era obseso sexual y tenía su diario grabado en un disco con un sistema de protección especial.

Otro caso es **Bill Landreth,** que después de ser arrestado y juzgado por introducirse ilegalmente en el correo electrónico «GTEmail», ha publicado un libro donde indica medidas para protegerse de sus excongéneres.

El Craker

El caso de este chico es quizás uno de los más conocidos. Sus inicios fueron casi casuales, ya que el primer intento lo hizo con un *modem* prestado y con el número de teléfono de un ordenador que le había dado un amigo. Al conectarse el ordenador le pidió el nombre del usuario, el respondió DAN, a continuación le pidió la clave de acceso y volvió a darle DAN. Entonces la máquina le dijo que no era válida y le rechazó. Una segunda prueba la realizó con

●●● mitiendo realizar eso. Además, al cabo de unos instantes se recibe otro mensaje del desconocido.

/ VENIMOS EN SON DE PAZ SOLO QUEREMOS HABLAR.

> MSG "NO TENEIS PERMISO PARA UTILIZAR ESTE ORDENADOR, DESCONECTAROS INMEDIATAMENTE SI NO QUEREIS SER DETENIDOS".

/ NO SABEIS QUIENES SOMOS NI DONDE ESTAMOS, ADEMAS NO PODEIS RASTREAR LA LLAMADA TELEFONICA, NO HEMOS CONECTADO POR LA RED TELEFONICA BANKLINK. NO NOS ENFADEIS.

El encargado del sistema adquiere ciertos tintes violáceos y después de pronunciar algunos insultos irrepetibles teclea en la consola.

> LOGON SYSMAX ********

Esta es una cuenta especial que permite hacer absolutamente todo, y su utilización es restringida debido a su peligrosidad. Pero antes de que puede teclear algo le llega otro mensaje.

/ SI NOS ECHAIS LO PASAREIS MUY MAL.

Pero nuestro hombre hace caso omiso y teclea.

> FORCE TEST, CONT1 LOGOFF: KILL ACOUNT CONT1, TEST

Antes de que el ordenador lleve a cabo la misión que le han dado de destruir esas dos cuentas y lo que estén haciendo, con lo que envía a los intrusos a la calle, les llega un último mensaje aterrorizante:

/ YA NOS VAMOS, PERO YO QUE VOSOTROS, REVISARIA EL BALANCE, PUEDE QUE ESTE MAL.

JIN, obteniendo resultados idénticos. Después de esto decidió olvidarse del tema, pero antes hizo un último intento con LEE, cuando el ordenador le pidió la clave de acceso y volvió a teclear LEE. Su sorpresa no tuvo límites cuando recibió como respuesta el mensaje «CLAVE VALIDA, BIENVENIDO AL SISTEMA». Con ello empezó a introducirse en este mundo. Al poco tiempo empezó a leer los BBS y tomar contacto con otras personas que se autodenominaban *hackers.* En conversaciones electrónicas con ellos aprendió sus primeros rudimentos, cómo entran en determinados sistemas, cómo conseguir aumentar el privilegio de su cuenta (que el ordenador le permitía realizar más cosas), etc. Otro apartado interesante era el intercambio de información referente a ordenadores, números de teléfono, claves de acceso, etcétera. Naturalmente, eso no lo consiguió de golpe. Todas estas comunicaciones se hacían por medio de BBS sin verse personalmente y utilizando seudónimos, el suyo era «el *cracker*» y tuvo que ganarse primero la confianza de los demás proporcionando información interesante de modo que todos pudieron ver que era de fiar.

Poco a poco su grupo de conocidos empezó a crecer y empezó a hacer correrías con algunos de ellos. Cierto día, uno apodado «Alpha Hacker» le propuso crear un grupo de gente de fiar de modo que la información entre ellos pudiese circular más libremente y con más seguridad. De resultas de ello surgió el *«Inner Circle»* grupo selecto de «bucaneros» con sus propias normas de conducta. En su decálogo existían normas muy estrictas que todos seguían a rajatabla, expulsando a los que nos las cumplían. Por ejemplo, tenían prohibido estropear cualquier información de un ordenador que no fuese



suya, con lo que conseguían pasar desapercibidos y si los descubrían, que el operador del sistema no se enfadase mucho con ellos.

Este punto era importante, ya que las reacciones de estas personas encargadas de velar por el funcionamiento del ordenador era diversa.

Estos acuerdos llegaban incluso hasta el nivel de que el *«Inner Circle»* velaba que no se introdujesen otras personas en el sistema, liberando al operador de mucho trabajo de vigilancia.

Durante su largo periplo visitó más de 800 ordenadores a todo lo largo de Estados Unidos, conviertiéndose en uno de los más famosos *«bucaneros».* Su última aventura la realizó en la red electrónica de correo de la GTE. Esta es una empresa especializada en servicios electrónicos a sus clientes ya que tenía un red de ordenadores distribuida por todo Estados Unidos que se podía usar mediante suscripción.

Una vez que entraron en este sistema empezaron a descubrir que funcionaba de un modo distinto a los demás. Normalmente no había operador del sistema y nuncan tenían problemas para entrar. Esto hizo que muchos se envalentonasen y se tomasen confianzas increíbles en otra situación. Dejaban los nombres verdaderos en sitios visibles dentro de la memoria, se pasaban la dirección, etc. Pero la gota que colmó el vaso y provocó el fin de gran cantidad de *«bucaneros»* fue la carta que envió uno de ellos a **GTE** en la que le contaba el estado de sus sistemas y se ofrecía para terminar con ello si le contrataban. La **GTE** rápidamente tomó cartas en el asunto y durante algún tiempo agentes del FBI observaron sus conversaciones electrónicas y tomaron cuidadosamente nota de todo lo que se decía. Un día llamaron a la

puerta de su casa y le detuvieron a la vez que confiscaban todo su equipo de ordenadores que tenía montado. Este fue el fin de su historia y el de mucha otra gente.

Lo que puede hacer un «bucanero»

Hasta ahora hemos hablado mucho de la historia y de los sucesos de los *hackers* y de los bucaneros, pero mucha gente se preguntará por qué son tan peligrosos. En primer lugar, existen tantos tipos distintos como personas hay en el mundo y por tanto, no se pueden hacer divisiones específicas. Unos simplemente curiosean en el ordenador y luego se irán. Otros, los amantes del peligro, incitarán a los operadores del sistema para ver si les pueden coger. Existen además los que se intentan aprovechar en beneficio propio pero sin pretender hundir a ninguna compañía. Como el protagonista de «Juegos de Guerra» que se cambiaba las notas y hacia uso del teléfono gratis.

Pero los más peligrosos son de otros dos tipos. Los primeros (que más que *hackers* o «bucaneros» son simples ladrones) pretenden obtener información que les pueda proporcionar dinero aun a costa de otras personas, como son números de tarjetas de crédito, etc., y por último están los que pretenden hacerse famosos y para ello destruyen toda la información contenida en el ordenador y dejando un mensaje como «esto lo hice yo».

Como resumen podemos decir que los *hackers* no son esos monstruos horribles devoradores de ordenadores que nos describen los periódicos. Más bien son personas agradables que comparten entre sí su gran afición por estas máquinas y disfrutan programando y descubriendo cosas nuevas.

GUN

¿Cómo se encuentran los POKES necesarios para modificar un juego? Analizamos un caso práctico: Gunfright.

Desde el primer momento en que se ve **Gunfright,** es inevitable compararlo con el anterior programa de **Ultimate Nightshade.** Al igual que en éste, se ha utilizado la técnica **Filmation II** consiguiendo la misma espectacularidad en los gráficos y movimiento. No obstante, el objetivo del juego ha variado sustancialmente. Nuestro héroe es un cazarrecompensas recién llegado a la ciudad que tiene que buscar una serie de pistoleros y enfrentarse con ellos en un duelo a muerte. Si consigue vencer cobrará la recompensa y así podrá ir amasando una fortuna, única finalidad del juego.





Mi primera partida fue un poco deprimente. Mientras se carga el programa, leo las instrucciones. Al acabar elijo la opción de teclado y empiezo a jugar con ideas poco claras sobre lo que hay que hacer. Mientras consigo hacerme con los mandos, pierdo un par de vidas en sendos choques con los ciudadanos que circulan por la ciudad. Ya en la última vida, empiezo a dar vueltas por el pueblo. Tras largo rato, hallo por casualidad un pistolero. Le disparo y me encuentro con él en un duelo en el que antes de darme cuenta estoy muerto.

En las siguientes partidas la situación no mejora mucho. Se tarda demasiado en localizar al pistolero y el duelo es tan rápido que no da tiempo a enterarse de cómo usar las teclas.

Esto no podía quedar así. Para conseguir averiguar lo que hay al final de este programa tenía dos caminos: dedicar muchas horas a jugar o entrar por la puerta de atrás y buscar todos los POKEs que facilitasen el objetivo. Evidentemente, elegí la segunda opción, mucho más apasionante.

Instalación del monitor

El primer paso fue instalar el programa en un *microdrive,*

FRIGHT



acompañado por un buen monitor, como, por ejemplo, el **MONS3M2.** En este punto aparecieron nuevas analogías con **Nightshade.** La protección es prácticamente idéntica y ambos comienzan en la dirección 5E00, aunque por problemas de espacio en memoria son cargados en 6000. A continuación elegí la dirección de memoria en la que iba a cargar el monitor. La 58000 viene bastante bien en esta ocasión, ya que es usada por el programa como área de trabajo para la construcción de los gráficos, con lo que podemos tener el monitor ahí, realizar cambios en el programa y luego lanzarlo. El único inconveniente es que no se puede volver al monitor, ya que es destrozado en el momento en que se lanza el programa. Por último, escribí un cargador que colocara todo el sistema en memoria. Ya estaba preparado para abordar el análisis del programa.

Por alguna extraña razón, el número de vidas iniciales es mayor del necesario

El bucle principal

Lo primero que se localiza es el bucle principal. No es muy largo, pero está lleno de llamadas a subrutinas. Curiosamente, un único bucle controla las tres fases del juego. Debe haber una variable que determine en cuál de ellas estamos.

El primer POKE que hay que buscar es el de vidas infinitas. El procedimiento consiste en tratar de encontrar en qué punto se inicializa una variable con el valor 2 ó 3, las vidas iniciales más las que estamos utilizando, y después ver cuándo se decrementa este dato. La asignación inicial de esta variable debe hacerse en alguna de las rutinas que son llamadas antes de entrar en el bucle principal. Sin embargo, buscando en esta zona no encuentro nada. Sólo en una ocasión se coloca un 4 como valor inicial de una variable. Este es un dato raro y pruebo a cambiarlo por un 5. Con el cambio, consigo una vida más. Por alguna extraña razón que luego descubriría, el número de vidas iniciales es mayor del necesario.

Buscando en qué punto se decrementa el número de vidas, encontré que en varias ocasiones es incrementado. Esto, en principio, parece indicar la posibilidad de conseguir vidas extras, y sin embargo, no es así. Pruebo a anular el punto en que se decrementa el número de vidas. La primera sorpresa al ponerme a jugar es que ahora cuento con cinco vidas. Me dejo matar y el número de vidas no disminuye. Todo parece ir bien. Encuentro al pistolero, le

disparo, me mata en el duelo y el programa se destruye. Algo ha funcionado mal. Analizando el programa descubro la explicación: al pasar del duelo rápido al pueblo, el número de vidas es incrementado y luego decrementado. Al haber anulado el punto en que se decrementa, el número de vidas crece ilimitadamente y al pasar de 5 el programa se hace un lío y se destruye. Existen dos opciones para solucionar el problema: anular todos los puntos en que se modifican las vidas o volver a inicializarlas cada vez que deban ser reducidas. Utilicé la última opción.

Con esto ya se pueden jugar partidas infinitas sin que nos maten, pero sigue siendo problemático encontrar al pistolero, sobre todo cuando descubro que algunos van a caballo y es difícil alcanzarlos. El siguiente objetivo es lograr saber en todo momento dónde está el pistolero y dónde estamos nosotros. Volvemos a la lucha con el código máquina.

Analizando la rutina donde se nos da a elegir entre las distintas opciones, aprendo a usar las subrutinas de impresión en pantalla. Por otro lado, el interferir el bucle principal para que ejecute continuamente nuestra propia rutina no presenta demasiados problemas. Lo único que queda por localizar es la dirección en la cual se guarda nuestra posición dentro del mapa y la del pistolero.

Dentro del principal, existe un bucle que se ejecuta 15 veces y en el que se va variando el valor del registro IX y llamando a una rutina distancia en función del dato (IX+0). Esto corresponde con los quince objetos en movimiento que el programa puede controlar. La información de cada objeto se guarda en 16 *bytes*, en los que se codifica el tipo de objeto, la pantalla en la que está, la posición dentro de la pantalla, la dirección en la que se mueve, la velocidad, etc. Esta información puede referirse a los ciudadanos, las tres balas, el pistolero, nosotros mismos o cualquier otra cosa que se pueda mover.

La información de cada objeto se guarda en 16 bytes, en los que se codifica desde la pantalla en la que están hasta la velocidad

Después de varias pruebas, logro descubrir cuáles de los 15 bloques son los que nos interesan, y también que los datos de la pantalla en que se encuentra corresponden a (IX+2) e (IX+4). Con todo esto ya se puede escribir la rutina que nos informa en todo momento de las posiciones en el mapa.

Esto facilita bastante el conseguir avanzar en el programa y enfrentarnos con los 20 pistoleros distintos con que cuenta (después se van repitiendo los cuatro últimos). De todas formas, los últimos se mueven muy rápido y es difícil toparse con ellos. Además, todavía hay que sufrir el duelo rápido, igual que al principio, con grandes probabilidades de perder. Quedan muchos POKEs que encontrar. Uno bastante interesante es el que evita que se mueva el pistolero. Con este objetivo me vuelvo a sumergir en el programa. Tras una búsqueda en las rutinas que controlan cada uno de los objetos móviles, descubro una rutina que desplaza un objeto en función del dato que se le pasa en el acumulador. El primer inten-

PROGRAMA 1

```
  10 LOAD ""CODE : LOAD ""CODE :
 PAPER 0: INK 0: LOAD ""CODE : R
ANDOMIZE USR 24576: PRINT AT 18,
0;: LOAD ""CODE : LOAD ""CODE
  20 POKE 23446,201: RANDOMIZE U
SR 23424: POKE 23446,33
  30 POKE 51544,233: POKE 42355,
0
 100 REM Comienzo de los POKEs
1000 RANDOMIZE USR 23446
```

PROGRAMA 2

```
  10 LET n=61440: RESTORE
  30 FOR i=1000 TO 1140 STEP 10
  40 READ a$,a: LET s=0
  50 FOR j=1 TO LEN a$-1 STEP 2
  60 LET d=16*(CODE a$(j)-48-7*(
a$(j)>"9"))+CODE a$(j+1)-48-7*(a
$(j+1)>"9")
  70 POKE n,d: LET n=n+1: LET s=
s+d: NEXT j
  80 IF s<>a THEN  PRINT "Error
en la linea ";i: STOP
  90 NEXT i
 100 RANDOMIZE USR 61440
1000 DATA "DD2100481100083EFF37C
D560530F1DD2100581100033EFF37CD5
60530F1DD21",2881
1010 DATA "0060110C1B3EFF37CD560
530F1CD0060DD2100601100803EFF37C
```


ORDENADORES
- QL - AMSTRAD - SPECTRUM

PROGRAMAS
- Contabilidad QL .. **20.000** ptas.
- Nóminas QL ..... **25.000** ptas.



TABLA 1

POKE 49746,54:	Vidas infinitas
POKE 49747,2:	
POKE 49748,0:	
POKE 43387,201:	Todo gratis
POKE 43927,0:	Caballo cerca
POKE 43168,X:	Número de bolsas iniciales
POKE 48528,X:	Número de bolsas en bonos
POKE 49772,X:	Velocidad
POKE 51997,X:	
POKE 47654,167:	Pueblo desierto
POKE 47427,0:	Los ciudadanos no mueren por las balas
POKE 47437,24:	Se puede atropellar
POKE 48116,24:	Pistolero parado en duelo rápido
POKE 48121,195:	Desenfunda en seguida
POKE 48121,33:	No desenfunda
POKE 51588,201:	Las balas se clavan en la pared
POKE 46570,62:	El pistolero no se mueve por el pueblo
POKE 46571,0:	
POKE 46572,0:	
POKE 47383,62:	Los ciudadanos no se mueven
POKE 47384,0:	
POKE 47385,0:	
POKE 52415,62:	Las balas quedan colgadas en el aire
POKE 52416,0:	
POKE 52417,0	
POKE 46555,24:	El pistolero es inmortal

to consiste en colocar una instrucción RET al principio de una rutina. El resultado no es muy satisfactorio, el pistolero probablemente no se moverá, pero no lo puedo comprobar porque yo mismo no me puedo mover. La rutina es llamada cuatro veces en el programa y anulando cada una de ellas por separado se consiguen resultados mucho mejores. La primera de ellas congela al pistolero, la segunda a los ciudadanos, la tercera deja las balas colgando en el aire esperando a que alguien se tropiece con ellas, y, por último, la cuarta llamada nos congela a nosotros mismos.

Un POKE para los ahorradores

Mucho antes de todo esto, había encontrado un POKE bastante útil si queremos conseguir muchos puntos. Con él, se logra que todo sea gratis. Lo encontré porque está en una rutina que se llama durante la inicialización del programa. También en esa zona encontré otro POKE que permite controlar el número de bolsas de dinero que aparecen en la primera fase, y ya puesto sobre la pista busqué el del resto de los bonos.

```
D560530F1DD",3083
1020 DATA "21805B1128003EFF37CD5
60530F121A15B36C9CD805BAF3273A32
158C736E921",3373
1030 DATA "005B22E3A32179F011005
B016401EDB0214C5B2291A3C3005E3A6
7A2A720243E",2983
1040 DATA "460801B08E213D503A7DA
2CDC2D23A7FA2CDC2D2217D503ABDA2C
DC2D23ABFA2",4148
1050 DATA "CDC2D23EF7DBFE1FD27CA
31F38053E01320AA21F380CDD2100401
1001B3EFFCD",3279
1060 DATA "C604C3BBC0CD75AB1E00C
DCCD62102403E43F5CDE1ADF13CFE473
8F62E622BF2",4352
1070 DATA "2152C036FA2336A52336C
0E5212A4011ED5BCDD0D4CD325CE1380
836002B3602",3273
1080 DATA "2B3636217BA73670E521C
A4011FF5BCDD0D4CD325CE1380236C93
E0A326CC032",3513
1090 DATA "1DC9216A48110C5CCDD0D
4CD325C38083E15326CC0321DC9214DB
73628E5210A",2970
1100 DATA "5011175CCDD0D4CD325CE
13802361821F9B936CAE5218A5011235
CCDD0D4CD32",3772
1110 DATA "5CE1D83621C9475649444
15320494C494D4954414441530046544
F444F204752",2601
1120 DATA "4154495300454143454C4
55241444F00444154524F50454C4C415
200434E4F20",2096
1130 DATA "44455345E46554E44410
0AFDBFE2FE61F20F83E7FDBFE2FE61F2
80811605CCD",3397
1140 DATA "D3D437C93EFDDBFE2FE61
F28E6115C5CCD9BAAA7C920534900204
E4F00",3521
```

Otra rutina interesante para nuestros POKEs, que fue descubierta por casualidad, es la que controla si un objeto está cerca de nosotros (o por lo menos eso creo que hace). Anulándola, conseguí que la ciudad estuviera desierta; nos quedamos a solas con el pistolero, pero no somos inmortales frente a él, como cabría esperar.

El control de la velocidad a la que nos movemos lo lleva una variable que es modificada tres veces en el programa: la primera cuando empieza la partida, la segunda cuando cogemos un caballo y la última cuando lo dejamos. Los valores normales son 0A cuando vamos a pie y 12 a caballo. Sin embargo, podemos darle cualquier otro valor, aunque los superiores a 17 harán que seamos tan rápidos que atravesaremos las paredes sin enterarnos, con lo que acabaremos saliéndonos del mapa, lo que provocará una progresiva degeneración de los gráficos que acabará por destruir totalmente el programa.

El otro punto del programa se diferencia si vamos a pie o a caballo, para saber si al atropellar a alguien perdemos la vida o es él quien la pierde. POKEando aquí, conseguiremos sobrevivir siempre a los encontronazos.

Por último, queda el problema del duelo rápido. La forma de localizar los POKEs fue a través de las variables que controlan en qué fase del programa estamos. Una vez localizada la rutina de movimiento del pistolero, vi que realizaba un test de



El control de la velocidad lo lleva una variable que es modificada tres veces en el programa

PROGRAMA 3

```
  10 CLEAR 25000: LET n=25025: R
ESTORE
  30 FOR i=1000 TO 1130 STEP 10
  40 READ a$
  50 FOR j=1 TO LEN a$-1 STEP 2
  60 LET d=16*(CODE a$(j)-48-7*(
a$(j)>"9"))+CODE a$(j+1)-48-7*(a
$(j+1)>"9")
  70 POKE n,d: LET n=n+1: NEXT j
  90 NEXT i
 100 SAVE "MAPA"CODE 25025,224:
STOP
1000 DATA "01000B000001010101010
10101010101"
1010 DATA "010002111901010101010
10101010101"
1020 DATA "010000000001010101010
10101010101"
1030 DATA "010C0A001001010101010
10101010101"
1040 DATA "010101010101010101010
10101010101"
1050 DATA "010101010101010101010
10101010101"
1060 DATA "010101010101010101010
10101010101"
1070 DATA "010101010101010101010
10101010101"
1080 DATA "010101010101010101010
10101010101"
1090 DATA "010101010101010101010
10101010101"
1100 DATA "010101010101010101010
10101010101"
1110 DATA "010101010101010101010
10101010101"
1120 DATA "010101010101010101010
10101010101"
1130 DATA "010101010101010101010
10101010101"
```

dos de sus parámetros y después un salto condicional. Sólo había que probar para qué servía ese salto. El primer intento consistió en hacer que fuese incondicional. De esta forma el pistolero desenfunda en el mismo momento en que se encuentra frente a nosotros. Resulta curioso como entrenamiento. Lo siguiente era anular el salto, y con esto se consigue que el pistolero no desenfunde nunca, únicamente se mueve.

Todos estos POKEs los tenéis en la tabla 1. Para utilizarlos tenéis que teclear el programa 1 introduciendo entre las líneas 100 y 1000 los POKEs que queráis. El programa 2 es un cargador en el que se han incluido los POKEs más útiles, la rutina que nos da nuestra posición y del pistolero, y tres teclas que nos permiten distintas opciones:

1. Aborta la partida.
2. Mata al pistolero aunque no esté en la pantalla.
3. Saca una copia de pantalla en casete.

Las copias de pantalla se sacan sin cabecera, para no alargar el programa. Para cargarlas se utiliza la cabecera de cualquier pantalla.

El mapa

Sólo queda por comentar la forma en que logré encontrar el mapa y cómo está codificado. Durante la búsqueda de los POKE's no había encontrado ninguna referencia a él. Pensando en el parecido con **Nightshade** intenté buscarlo en la misma dirección, pero no estaba allí. Sería demasiado fácil. Sin embargo, sí estaba codificado de la misma forma, y eso fue lo que me dio pie para encontrarlo. El mapa está almacenado por columnas, con un *byte* por pantalla que se puede codificar de varias formas, esto no es así, puesto que los gráficos siempre son distintos. El total hay 16*16 pantallas, pero las filas y columnas primera y última están llenas de 01 para marcar los límites del mapa, con lo que queda reducido a 14*14. Una vez descubierto esto, introducir un nuevo mapa no presenta demasiados problemas. El programa 3 es un ejemplo de ello. Cada línea de DATAs representa una columna del mapa (quitando la primera y última). Para introducir el mapa, ejecuta el programa 3 y guarda en una cinta el código resultante. Luego carga el programa 1 y pone en la zona de POKEs la línea LOAD "MAPA" CODE. Por último, carga el programa y después el mapa. El de este ejemplo se ha hecho de 4*4 para que sea más fácil encontrar al pistolero.

Manuel Alana

Mapa Original
Codificación de Pantallas y nuevo mapa del hacker.

NOTI



Challenger: la NASA sustituirá sus ordenadores en 1987

Los ordenadores que controlan la lanzadera espacial, construidos por IBM a mediados de los 70, han quedado desfasados en comparación con los actuales.
Para aumentar la seguridad del transbordador espacial, cuatro de las cinco máquinas de a bordo trabajan en un sistema operativo de IBM, mientras la quinta utiliza un sistema con las mismas especificaciones pero escrito por otra empresa.

CP/M PARA SPECTRUM

Timex Portugal se dispone a lanzar próximamente un completo sistema de disco destinado al Spectrum. Constituido por dos *drives* para discos de 3 pulgadas (formato utilizado por Amstrad), un microprocesador Z-80 y dos *ports* de impresora, incluye el conocido sistema operativo CP/M.
Los proyectos de Timex no acaban ahí, pues tienen prevista la comercialización de un teclado profesional que, conectado a la unidad de discos, reemplaza al Spectrum y convierte al conjunto en un nuevo ordenador.
Como comentábamos el mes pasado, Sinclair también está trabajando para incorporar CP/M al Spectrum, aunque afirma que no estará disponible hasta el lanzamiento de su ordenador portátil.



Sinclair vende la televisión de bolsillo a Timex

Los derechos de comercialización y distribución de la televisión de bolsillo de Sinclair han sido adquiridos por Timex, empresa que se encargó de su construcción desde el momento de su lanzamiento.
No obstante, Sinclair Research continuará investigando en su cuartel general de Cambridge la tecnología de la pantalla plana.

C









SINCLAIR CONSERVA EL LIDERAZGO DEL MERCADO

Durante 1985, Sinclair se mantuvo como líder indiscutible del mercado británico de ordenadores domésticos, con un porcentaje del 37 por 100 sobre el total de ventas. Por debajo se encuentran Commodore, Amstrad y Acorn, con cifras muy inferiores.

Club de usuarios de QL

A finales del pasado año se constituyó en Zaragoza el «Club español independiente de usuarios del QL» (QLAVE), con el propósito de servir como medio de unión, información y comunicación de los usuarios de QL españoles e hispano-hablantes. Quienes estén interesados en ponerse en contacto con QLAVE pueden dirigirse a:
Serafín Olcoz Yanguas
Baltasar Gracián, 21, 1 Cto.
50005 Zaragoza
También en Madrid existen iniciativas encaminadas a la creación de un club de usuarios de QL del que esperamos poder ofrecer más información en próximos números.

¿Qué pasa con el Atari 520 ST?



Nunca una máquina con unas críticas iniciales tan favorables ha pasado tan desapercibida como el Atari 520 ST. En nuestro país apenas se ha oído hablar de él tras sus breves apariciones en el Sonimag y el SIMO. Tampoco Investrónica, su distribuidor oficial, hace demasiado para darlo a conocer.
De los dos programas que en principio estaba previsto entregar junto a la máquina, Gem Write y Gem Draw, nunca más se supo. Al parecer, han sido sustituidos por First Word, un procesador de textos de GST (empresa que ha realizado algunos programas para el QL) y DB Master One, base de datos de DB Master.
El sistema operativo TOS y el GEM, que, según el fabricante se incluirían en memoria ROM, se sigue entregando en diskette y al parecer todavía con abundantes bugs.
El problema que afecta al 520 ST parece el mismo que estuvo a punto de acabar con el QL hace apenas unos años: lanzamiento prematuro y escasez de software.

JUEGOS



COMMANDO

ZAFIRO

SPECTRUM 48 K

Aunque el tema no es original, pues el cine lo ha explotado ya hasta la saciedad en películas como Rambo, Acorralado y el mismísimo Comando, el juego es sencillamente sensacional. Quien coja el *joystick* y se siente frente al televisor comprenderá rápidamente el alcance de esta afirmación cuando se dé cuenta de las horas que Commando le ha enganchado a su silla.



Sus gráficos, color, movimiento y sonido aprovechan al máximo las características del Spectrum, logrando importantes cotas de adicción al colocarte continuamente en situaciones límites. Sólo tu habilidad y destreza en el manejo del *joystick* conseguirán sacarte de apuros durante los enfrentamientos con las fuerzas enemigas.

Para completar la misión cuentas con un fusil ametrallador y bombas de mano, de gran utilidad frente a los enemigos parapetados en trincheras y los vehículos blindados. Las balas a tu disposición son ilimitadas, pero no así las granadas, por lo que tendrás que recoger las que encuentres en el camino, procurando no quedarte en ningún momento sin ellas.

En la línea de los clásicos juegos de arcade, en los que todo es acción, Commando es indudablemente uno de los mejores programas de Elite y también de Zafiro, la empresa que lo distribuye en España.

Si te encuentras preparado para tu bautismo de fuego, ¡suerte en el combate! y... ¡al ataque!

SWEEVO'S WORLD

GARGOYLE GAMES

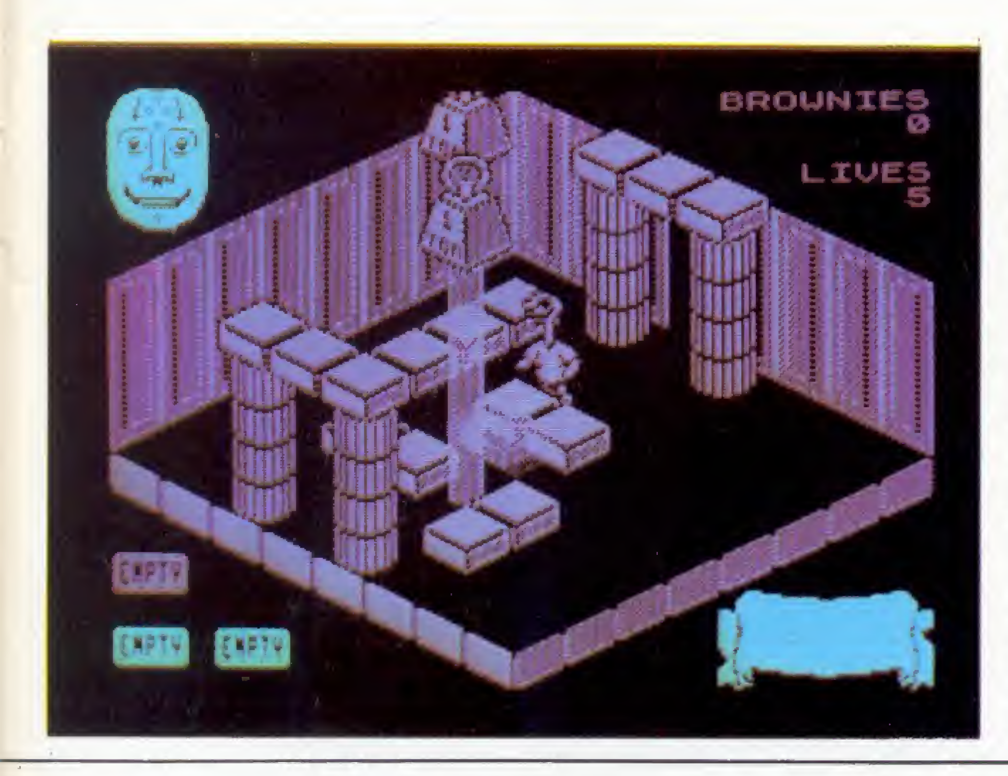
SPECTRUM 48 K

Conocida por sus brillantes juegos de aventuras, como Tir Na Nog, Dun Darach y Marsport, Gargoyle Games entra en el terreno de los juegos de arcade con Sweevo's World.

Caracterizado por una imaginación sin límites en cuanto a las situaciones y personajes que aparecen en la pantalla, los gráficos tridimensionales y la animación de Sweevo's World son muy parecidos a los Knight Lore y otros programas de Ultimate realizados con la técnica Filmation.

El protagonista de la historia, el robot Sweevo, ha de viajar por los cuatro mundos que componen el juego recogiendo diversos objetos. Aunque en la pantalla de carga aparece la imagen de Stan Laurel «el flaco», Sweevo no se parece de-





masiado al famoso actor cómico americano, sino más bien a ET.

Existen cuatro puntos de partida para la aventura, seleccionables por el jugador, que corresponden a otros tantos mundos diferentes: Totalmente Libre, Pino Solitario, Dedos y Tarta de Manzana, a los que se llega tras un descenso en paraguas. A partir de ese instante, viviremos las situaciones más sorprendentes que podamos imaginar. Las numerosas pantallas que forman el juego hacen imprescindible la utilización de un buen mapa.

Comparable a los mejores juegos de Ultimate, Sweevo's World es un magnífico programa que desgraciadamente, al menos que nosotros sepamos, todavía nadie distribuye en España.

Cuide su Spectrum



Proteja su ordenador y manténgalo como nuevo con esta práctica funda de teclado transparente

Servicio especial para nuestros lectores y amigos



RECORTE Y ENVIE **HOY MISMO** ESTE CUPON A:
PUBLINFORMATICA, C/BRAVO MURILLO, 377 5.° A 28020 MADRID

CUPON DE PEDIDO

SI envieme al precio de **950** Ptas. cada una ______ fundas para mi SPECTRUM

El importe lo abonaré: Con mi tarjeta de crédito ☐ American Express ☐ Visa ☐ Interbank ☐

Contra reembolso ☐ Adjunto cheque ☐

Numero de mi tarjeta ______

Fecha de caducidad ______

NOMBRE ______
DIRECCION ______
CIUDAD ______ C.P. ______
PROVINCIA ______

Sin gastos de envio

APROVECHE ESTA OPORTUNIDAD Y BENEFICIESE DE UN 30 % DE DESCUENTO SOBRE SU PRECIO NORMAL DE VENTA

SGRIZAM

DINAMIC

SPECTRUM 48 K



Dinamic nos presenta SGRIZAM, una increíble aventura futurista que nos traslada al siglo XXV. En ella tenemos dos grandes objetivos que alcanzar: el primero recuperar las riquezas del palacio imperial que fueron robadas por los guerros del planeta Kindos y el segundo rescatar a la joven princesa Doxaphin. Si conseguimos los dos objetivos, el Emperador nos ha prometido la mano de la princesa y la mitad del tesoro robado. Sólo disponemos de tres vidas y la mejor espada del imperio, SGRIZAM.

Nuestro guerrero se adentrará en el castillo de los Kindos, donde deberá esquivar toda clase de animales y objetos que se interpongan en su camino, ya sea saltando sobre ellos o agachándose. También encontrará guerreros guardando el castillo, así que tendrá que combatir contra ellos utilizando la gran espada. Cuando haya recorrido una parte del largo pasadizo, pasará a la fase siguiente. La dificultad aumenta con las diferentes frases y, especialmente al alcanzar el primer objetivo.

Los gráficos son sin duda la principal atracción del juego, aunque no dan demasiada importancia al decorado y se centran principalmente en la definición de los guerreros. Los movimientos son muy elementales.



PANZADROME

ARIOLA SOFT

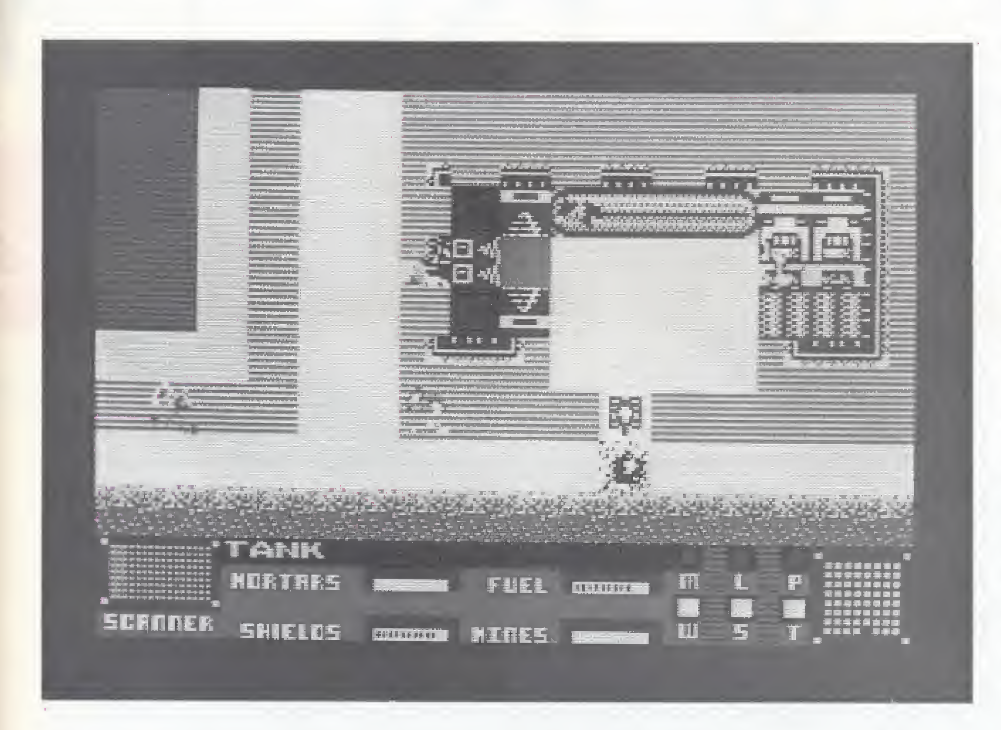
SPECTRUM 48 K

Conducir un tanque indudablemente no es tarea fácil, y la cosa se complica aún más cuando debes enfrentarte en el laberinto de la ciudad con una nutrida muestra de elementos bélicos que acosan, atacan y destruyen tu vehículo.

Baterías de artillería disparando a mansalva, tanques apareciendo por todas las direcciones y calles estrechas son las mayores dificultades que podemos encontrarnos.

El juego comienza con una pantalla de presentación curiosa, pero nada brillante, donde se muestran las distintas partes del vehículo y el armamento que posee como dotación.

Con este armamento, tendrás que enfrentarte a las dificultades ya mencionadas, partiendo desde tu base.



Mucho cuidado con los disparos enemigos en tu recorrido. Observarás que si te disparan y no aciertan, producen una explosión en el suelo por donde no es posible pasar, pudiendo en ocasiones quedar cercados.

Desde un punto de vista técnico, si bien los gráficos de la ciudad y el movimiento son dignos, no aportan nada extraordinario al *software* que existe en el momento, quedándose por debajo del refinamiento de los programas actuales, tanto bélicos como laberínticos, donde las tres dimensiones son condición necesaria en un programa de calidad, hecho que en este caso no se produce.

Tampoco el sonido es fuerte, así como deja bastante que desear la presentación e instrucciones adicionales, que por descontado se encuentran en inglés, todo un detalle ahora que por fin «somos europeos».

En resumen un programa normalito cuyo adjetivo más apropiado sería el de vulgar.



SUSCRIBASE POR TELEFONO

- más fácil,
- más cómodo,
- más rápido

Telf. (91) 733 79 69

7 días por semana, 24 horas a su servicio

SUSCRIBASE A **Todospectrum**

APREN

LENGUAJ

Un microprocesador que no sea capaz de comunicarse con el exterior es como un campanario sin cigüeña: no tiene razón de ser. ¿De qué nos serviría un ordenador sin teclado, sin monitor, sin altavoz y sin sistemas de almacenamiento de datos? Evidentemente para nada. Es por ello por lo que todos los procesadores tienen reservado algún espacio de direcciones para dispositivos de entrada y salida. Pero, si las 64K que es capaz de direccionar el Z-80 son utilizadas en su totalidad (al menos en el 48K) entre ROM y RAM, ¿cómo se las arregla el Spectrum para funcionar?

Existe una pista en el *bus* de control (IORQ) que indica cuándo el microprocesador está realizando una operación de entrada y salida y cuándo no. Por lo tanto, cuando IORQ esté activado, podremos disponer del *bus* de direcciones, es decir, de 64K extras, para dedicarlas exclusivamente a entradas y salidas.

En lugar de asignar a cada dispositivo una o varias direcciones, para mayor versatilidad del sistema, en el Spectrum cada *bit* de una dirección indica si un dispositivo está seleccionado (valor 0) o no (valor 1).

Así, el *bit* 0 de una dirección indica si se utiliza la ULA, el 2 si la ZX-Printer, y el 3 y el 4 si el Interface 1 y los microdrives. Como podemos ver quedan los *bits* 1 y 5-15 para otros periféricos, aunque, en lo que atañe a entradas, los *bits* 8-15, es decir, el *byte* más significativo, se utilizan para el teclado como se explica más adelante.

Los INs y los OUTs

Existen dos instrucciones del ensamblador (también presentes en el BASIC), que se encargan de efectuar entradas y salidas. Equivalen a PEEK y POKE, sólo que activan IOQR, por lo que el dato que «leamos» o «escribamos» no irá/vendrá a/de memoria, sino a/de el dispositivo que corresponde a la dirección seleccionada o *port*.

La sintaxis es la siguiente:

IN A, (n) lee en el acumulador el *port* direccionado por el propio valor de A como *byte* alto y el número n como *byte* bajo.

IN r, (C) lee en r (= A, B, C, D, E, H o L) el *port* direccionado por BC.

OUT (n), A saca el número n por el *port* direccionado por A como *byte* alto y n como *byte* bajo.

OUT (C), r saca el valor de r (= A, B, C, D, E, H o L) por el *port* direccionado por BC.

Al igual que con las instruccio-

DIENDO

Capítulo 7

E MAQUINA

nes de carga y comparación, existen instrucciones de entrada y salida que incrementan o decrementan y repiten.

INI, en *assembler,* no es ninguna agrupación de industrias en malas condiciones, sino una instrucción que lee en la posición de memoria direccionada por HL el *port* direccionado por BC, incrementa HL y decrementa B. INIR realiza un INI pero lo repite hasta que B valga cero. Similares a las anteriores, IND e INDR, en lugar de incrementar, decrementan el par HL.

OUTI saca por el *port* direccionado por BC el valor de la posición a la que apunta HL, incrementa HL y decrementa B. OTIR repite hasta que B sea igual a cero. OUTD y OTDR igual, pero decrementando HL.

Como vemos, estas últimas instrucciones modifican el *byte* de mayor peso a medida que ejecutan, por lo que sólo son útiles cuando éste no es utilizado. INIR y OTIR son usadas, por ejemplo, para efectuar la lectura y escritura de los sectores en los microdrives.

La ULA, un ayudante eficaz

La ULA es un *chip* diseñado especialmente para ayudar al Z-80 en tareas específicas del Spectrum. Controla, por tanto, todas las entradas y salidas que puede efectuar un Spectrum *standard* sin otros interfaces.

La ULA se comporta de distinta forma según la utilicemos como dispositivo de entrada o de salida. En todo caso, para seleccionarla, el valor que la direccione tendrá que tener a cero el bit 0.

Como dispositivo de salida, controla el color del borde (*bits* 0-2), la toma MIC (*bit* 3) y el altavoz (*bit* 4). Los *bits* 5-15 no son utilizados en este caso. La dirección que habremos de usar para estos dispositivos es la 254 o 11111110 bin (es indiferente el valor del *byte* más significativo).

Por ejemplo, OUT (254), A, siendo A igual a 2, pondrá rojo el color del borde, aunque si no actuamos sobre la respectiva variable del sistema volverá a su estado inicial en cuanto ejecutemos alguna operación del BASIC que le afecte.

La Figura 1 muestra el listado de una rutina que cambia el color del borde a gran velocidad y testea, mediante la subrutina de la ROM BREAK-KEY (1F54hex), si BREAK está siendo pulsado, para volver al BASIC sólo en ese caso. La velocidad a que cambia de color es tal que el ojo humano no es capaz de seguirlo, por lo que el resultado es un borde multicolor y estático. Precisamente para que las líneas no suban o bajen se han incluido las instrucciones DI y EI, que prohíben las interrupciones durante la ejecución de la rutina. Esto no será suficiente si colocamos la rutina en las primeras 16K de RAM, por lo que los usuarios de Spectrum 16K observarán bastante movimiento mientras la ejecutan. La forma en que actúan las interrupciones será estudiada más

FIGURA 1

```
 10          ORG   60000
 20
 30          DI
 40 NO_BRK
 50          LD    B,7
 60 CAMB_B
 70          OUT   (C),B
 80          DJNZ  CAMB_B
 90          CALL  #1F54
100          JR    C,NO_BRK
110          EI
120          RET
```

FIGURA 2

```
 20          ORG   60000
 30
 40          DI
 50          LD    L,1
 60 N_FREC
 70          LD    H,10
 80          LD    A,(23624)
 90          RRA
100          RRA
110          RRA
120 M_FREC
130          XOR   16
140          OUT   (254),A
150          LD    B,L
160 RETARD
170          DJNZ  RETARD
180          DEC   H
190          JR    NZ,M_FREC
200          INC   L
210          JR    NZ,N_FREC
220          EI
230          RET
```

BUS DE DIRECCIONES

BUS DE DATOS

U. L. A.

FIGURA 4

extensamente en capítulos posteriores.

Haciendo ruido

Para hacer sonar el sencillo altavoz del Spectrum habremos de hacer un OUT a 254, activando y desactivando su *bit* 4 a gran velocidad (de ésta depende la frecuencia del sonido). Si nos limitamos a activarlo durante un período determinado de tiempo no conseguiremos más que un inaudible «click». Hay que tener cuidado de lo que hacemos con los *bits* 0-2, pues, si no usamos los valores apropiados, el color del borde cambiará a dúo con el altavoz.

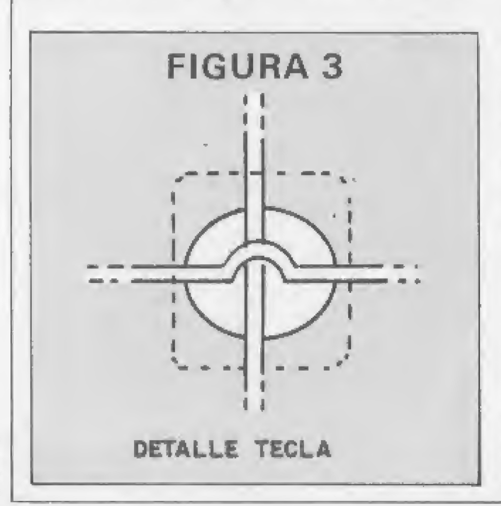


FIGURA 3

La Figura 2 muestra el listado de una rutina que hace sonar el altavoz de forma muy distinta al conocido BEEP del BASIC. Lo que hace especial al sonido generado por esta rutina es que la frecuencia va modificándose a medida que se ejecuta, por lo que el resultado es algo parecido a una sirena. Se puede variar la longitud y características del sonido modificando el valor de H en la línea 70; un número impar dará un sonido más estridente.

Obsérvese cómo esta rutina, mediante RRCA, toma de la variable del sistema BORDCR (23624) el color actual del borde para que no sea modificado. También usa XOR 16 para poner a 1 el *bit* 4 si estaba a 0 y ponerlo a 0 si estaba a 1 (el qué hacen y cómo se usan las instrucciones RRCA y XOR lo dejaremos para el capítulo siguiente, en el que veremos todas las instrucciones lógicas y de rotación). Como la rutina de la Fig. 1, prohíbe las interrupciones con DI al comienzo y las vuelve a permitir con EI antes de la vuelta al BASIC, esto es necesario para que el sonido sea puro, si no lo hiciéramos, las interrupciones de la ULA crearían un zumbido de fondo de 50Hz que echaría todo por tierra (esto sucederá también si ubicamos la rutina en las primeras 16K de RAM).

Si nos conformamos con hacer pitidos oficiales Sinclair, podemos recurrir a la subrutina de la ROM BEEPER, localizada en 03B5hex y a la que podremos pasarle en el par DE el producto de la frecuencia del sonido (en Hz) con su duración

(en segs), y en HL el número de estados temporales del bucle de retardo dividido entre 4. Este es igual a la frecuencia/2 por la velocidad de reloj (en el Spectrum 3.5MHz). Esta rutina, por estar en ROM, sí puede ser utilizada por los usuarios del 16K.

Usada como dispositivo de entrada, la ULA controla el teclado y la toma EAR. Esta última es seleccionada por el bit 6 de la dirección del *port*. En cuanto al teclado, usa los *bits* 8-15.

El teclado, otro periférico

Debemos ver el teclado del Spectrum como una malla en la que se cruzan las pistas del *byte* alto del *bus* de direcciones con las correspondientes a los bits 0-4 del *bus* de datos (ver Fig. 3). En los puntos de cruce hay un interruptor que corresponde a cada una de las teclas y puede conectar una pista con la otra. Este es el esquema utilizado en el Spectrum de 16 y 48K; en el Plus y el 128K la cosa se complica al unir algunos contactos más de dos pistas, pero el resultado final es el mismo. Así, según aparezcan en la dirección de un *port* los *bytes* 8-15 puestos a cero, podremos saber el estado de las semifilas que les corresponden en la malla. El resultado será un número cuyos *bits* 0-4 puestos a cero indicarán si están pulsadas cada una de las teclas de la semifila seleccionada. El valor de los *bits* 5-7 del resultado no es significativo, y de hecho cambia con las distintas versiones del Spectrum.

Dado lo sencillo del sistema utilizado, puede haber problemas de realimentación cuando se pulsan más de dos teclas a un mismo tiempo, por ejemplo, si pulsamos CAPS SHIFT, «x» y «m» a la vez cualquier sistema de lectura que empleemos creerá que también SPACE está siendo pulsada (puedes probar a utilizarlo como BREAK).

Para comprobar el estado de una determinada tecla o grupo de ellas podemos acudir a la tabla de la Figura 4, en la que se dan los distintos valores que corresponden a las



ENSAMBLADORES

EDITOR ENSAMBLADOR PICTURESQUE

Continuamos con los comentarios sobre los programas ensambladores más característicos de los presentes en nuestro mercado. En esta ocasión le toca el turno al **Editor Ensamblador Picturesque,** otro de los preferidos por los aficionados al código máquina por sus prestaciones y facilidad de manejo.

En líneas generales se trata de un ensamblador bastante completo. El editor de líneas es fácil de utilizar, su único defecto es que carece de la posibilidad de inserción, algo que sí tienen ensambladores de menos calidad. Salvo este detalle, ofrece casi todas las opciones que se le pueden pedir a un programa de este tipo: remuneración, auto, utilización con cualquier impresora, grabación y carga en *cassette* y *microdrive,* etcétera.

Admite números en base decimal y hexadecimal y para esta última hay que indicarlo con una «H» después del número. Si un número hexadecimal comienza por una letra hay que ponerle un cero delante para que el ensamblador lo distinga de las etiquetas. Es obligado el uso del comando END al final de cada listado. No admite que incluyamos las etiquetas en líneas independientes, por lo que en los listados que utilicen esta técnica habrá que fusionar cada una de esas líneas con la que le siga, por ejemplo:

```
LD B,  FO
RETARD
DJNZ RETARD
RET
```

deberá sustituirse por:

```
LD B, OFOH
RETARD DJNZ RETARD
RET
END
```

El programa no es reubicable, aunque incluye en el mismo paquete versiones para 16 y 48K (algo recortada y por tanto más limitada la primera). Esto no es en realidad un problema grave, dado el uso que hace de la memoria, en el que el programa fuente crece hacia abajo en el espacio de reserva del BASIC.

En definitiva, se trata de un ensamblador bastante cómodo de utilizar por el uso de palabras-clave al estilo del BASIC que acorta el período de familiarización con el programa. Recomendado para quien empieza.

ocho semifilas del teclado. Si sólo queremos saber si una tecla cualquiera ha sido pulsada, podemos utilizar el *port* 254 y comprobar si en los *bits* 0-4 del resultado hay algún cero.

En realidad el conocer todo esto sólo nos resultará de verdadera utilidad cuando deseemos saber si se están pulsando dos teclas a un tiempo. Para necesidades menos sofisticadas es más práctico utilizar las rutinas de la ROM como hicimos en el capítulo 5.

Otros dispositivos

Hay docenas de dispositivos distintos que pueden ser conectados al Spectrum. Esto se hace gracias a los *interfaces,* que, en realidad, no son sino dispositivos de entrada y salida que sirven de enlace entre el Z-80 y *microdrives,* impresoras, discos, *joysticks,* lápices ópticos, otros ordenadores y un largo número de etcéteras.

Uno de los más utilizados es el **Interface 1,** que permite el uso del RS232, la red local y los *microdrives.* La mejor manera de utilizar estos dispositivos es aprovechando las rutinas de la nueva ROM que incorpora el mencionado interface como veremos más adelante.

Otro mítico periférico del Spectrum es la **ZX-Printer,** que utiliza el *port* 251. Haciendo un OUT a esa dirección con el *bit* 7 activado imprimirá un punto, y si el *bit* 2 está a cero pondrá en marcha el motor. Algunos *interfaces* de impresora y otros dispositivos utilizan los *bits* 6 ó 7 para direccionar su *port,* o inclusive *bits* del *byte* alto mientras mantienen el de la ULA a 1.

Es posible que algunos de los términos utilizados en este o en alguno de los anteriores capítulos no hayan sido comprendidos del todo por los menos aventajados. No deben desesperar, aunque el principio puede ser realmente duro, a medida que se adentren en el estudio de esta bonita materia las cosas se irán haciendo más fáciles. Es entonces cuando se debe volver atrás y releer todo cuanto nos hizo dudar la primera vez.



LA PROTECCION DEL SOFTWARE

QUILL
ABACUS
EASEL
ARCHIVE
QL-Monitor
QL-Assembler
QL-Toolkit

ICE: Iconos, ventanas y ratones



La protección del software

La protección de programas tiene un significado distinto para cada uno de los diferentes tipos de usuarios existentes. En la escala superior, las casas de software emplean medidas de protección para mantener el fruto de costosas inversiones fuera del alcance de la explotación criminal de terceras personas (piratas). Para el programador, es un medio de mantener en secreto el resultado de muchas horas de duro trabajo desarrollando revolucionarios algoritmos. En cuanto al hombre de negocios, significa impedir el acceso a información reservada. Finalmente, puede ser también, para el simple aficionado, un medio eficiente de cobertura de una mala programación. En este artículo, vamos a revisar algunos métodos para proteger sus propios programas.

Debido a que la entrada y salida de datos bajo el sistema operativo QDOS es un dispositivo independiente, la protección de programas resulta algo problemática. Cualquier archivo, esté grabado en *diskette* o *microdrive*, puede copiarse fácilmente en una impresora, pantalla o cualquier otro *diskette* o *microdrive*. Esto es particularmente cierto para los programas SuperBASIC que se han almacenado como archivos en forma de caracteres ASCII, ya que el comando COPY mdvl_nombre_de_archivo TO scr revelará al instante toda la información de forma totalmente legible.

En cuanto al código máquina, aunque éste necesita desensamblarse, el hecho de que el QDOS permita extraer *bytes* de cualquier lugar de la memoria mediante el comando SBYTES y volverlos a cargar con LBYTES en otra localización totalmente diferente, facilita la desprotección.

Por lo tanto, para proteger un programa de QL deben idearse métodos especiales, para que solamente funcione cuanto esté en su soporte original o para hacer el proceso de copia lo más complejo posible. En este artículo pasaremos revista a algunos métodos disponibles para el usuario.

RUTINA DE CARGA AUTOMATICA

La carga de programas comerciales desde el soporte en que han sido grabados (cartuchos *microdrive*) a la memoria de modo que funcionen automáticamente, se realiza mediante un archivo BOOT.

En el QL, inmediatamente después de pulsar una de las teclas de función para seleccionar el formato de pantalla, la máquina intenta abrir un canal para un dispositivo llamado BOOT, consistente en una lista de comandos almacenados en ROM. Si falla el intento, se prueba a abrir un archivo de entrada BOOT que esté almacenado en el *microdrive* 1. Este archivo, formado por una serie de líneas BASIC en ASCII, se cargará normalmente en la mayoría de los programas SuperBASIC y/o código máquina (incluyendo una sentencia CALL para estos últimos).

Este mecanismo gira en torno a la variable del sistema BV_COMCH, en la dirección $0084(A6), la cual contiene una palabra de información con el canal ID para el comando de entrada de flujo. Normalmente dicho contenido es cero para el canal de entrada del teclado, pero al producirse una carga automática (BOOT) o al entrar en funcionamiento el comando LOAD, contendrá el canal ID para archivo abierto.

En lugar de seguir el procedimiento de carga automática descrito más arriba, el pirata podría introducir esta orden: LOAD mdv1_boot, e inmediatamente listar el programa para descubrir más información. Pero como este archivo es una lista de comandos que no necesitan números de línea, se autoejecutará siempre aunque se emplee LOAD o MERGE.

El listado 1 es una demostración de cómo establecer un archivo de ejecución automática (BOOT). Una vez grabado, intente cargar el programa con LOAD o MERGE y pulse BREAK a continuación (CTRL + SPACE). Seguidamente, introduzca el comando LIST y descubrirá que el programa ha desaparecido, ya que no es un programa propiamente dicho, sino una lista de comandos directos provenientes del cartucho de *microdrive*. Este método puede usarse para cualquier programa SuperBASIC (no necesita llamarse BOOT ni tampoco tiene por qué estar forzosamente almacenado en el *microdrive* 1).

INVISIBILIZANDO ARCHIVOS

Aunque el método anterior puede hacer desistir temporalmente al pirata, todavía puede estar en condiciones de utilizar el comando COPY. En este caso deberemos fabricar un archivo invisible, de



modo que no aparezca en el directorio. Un ejemplo de utilización de este método es SBYTES mdvl_,0,49152. En este caso, grabaríamos el contenido de la ROM en el *microdrive* 1 con un nombre de archivo nulo. Al efectuar un DIR mdvl_, el directorio no muestra ningún nombre de archivo, pero el número de sectores libres se verá sensiblemente reducido.

Como alternativa, también puede almacenar archivo con nombres como mdvl_«ESPACIO». Estos no pueden copiarse porque el QL no puede trabajar si tras el subrayado encuentra un caracter de espaciado.

DESACTIVACION DE BREAK

Uno de los métodos más conocidos para evitar la piratería consiste en inhabilitar la acción de BREAK.

El teclado sufre una exploración cada 20 milisegundos debido a las interrupciones generadas por una de las ULAs del QL. Afortunadamente, este mecanismo de interrupción está condicionado por el puntero de una tabla de datos situado en SV_PLIST (en la dirección $2803C). Esta palabra apunta a una lista de 8 *bytes*. Los primeros 4 *bytes* son el puntero del siguiente elemento de la tabla de datos. Los otros 4 *bytes* contienen el puntero del mecanismo de interrupción de 50 Hz que debe ejecutarse en ese momento. Al encenderse, el QL sin expansión sólo tiene una entrada en la tabla, consistente en una rutina (llamada cada vez que se produce una interrupción) que trabaja con todos los *interfaces* externos del QL mediante la comunicación con las ULAs principales y el IPC *(Inteligent Peripheral Controller)*.

Para desactivar BREAK desde el SuperBASIC solamente debe introducirse el siguiente comando: POKE_L 163900,0.

Al colocar un 0 en esa dirección de memoria, el QL ignora las tareas externas por estar vacía la tabla. Esto afectará a cualquier *hardware* extra que se esté utilizando y emplee interrupciones de 50 Hz.

LISTADO 1

```
130 OPEN_NEW #4,mdv1_boot
140 PRINT #4,"REPeat loop:";
150 PRINT #4,"  PAPER RND(0 TO 7):";
160 PRINT #4,"  PRINT 'Try pressing (BREAK)'::";
170 PRINT #4,"END REPeat loop"
180 CLOSE #4
```

LISTADO 2

```
MT_LSCHD       EQU      $1E             INTERRUPCION *1
BASIC_HEADER   EQU      $68             TAMANO CABECERA CARGADOR SB
BV_BRK         EQU      $00BF
JB_START       EQU      $0014
SV_BASIC       EQU      $28010          DIRECCION CABECERA SB
SV_SCRST       EQU      $28033          DIRECCION FLAG ESTADO PANT.
WAITING        EQU      -2

* COMIENZO DEL CODIGO

               LEA      TASK,A1         PUNTERO PARA LA NUEVA TAREA
                                        DE PLANIFICACION
               LEA      TASK_LINK,A0    PUNTERO PARA RESERVA DE ESP.
               MOVE.L   A1,4(A0)        ALMACENA PUNTERO DE TAREA
               MOVEQ    *MT_LSCHD,D0    INTERRUPCION
               TRAP     *1
               RTS                      RETORNO AL SB

* TAREA DE PLANIFICACION A LLEVAR A CABO

TASK           MOVEA.L  SV_BASIC,A0     ESTABLECE DIRECCION DE LA
                                        CABECERA DE LA RUTINA SB
               LEA      BASIC_HEADER(A0),A1
               TAS      BV_BRK(A1)
               TST.W    JB_STAT(A0)     EXAMINA ESTADO TAREA SB
               BGE.S    NO_ALTER        NO ALTERAR SI SUPERBASIC NO
                                        ESTA SUSPENDIDO
               MOVE.    *WAITING,JB_STAT(A0)
NO ALTER       CLR.B    SV_SCRST        SE ASEGURA QUE LA PANTALLA
                                        ESTE ACTIVA
               RTS

* RESERVA ESPACIO DE TRABAJO

               DS.L     2

               END
```

LISTADO 3

```
130 LET reserved_address=RESPR (60)
140 LET address=reserved_address
150 RESTORE
160 REPeat read_and_store_machine_code
170 IF EOF THEN EXIT read_and_store_machine_code
180 READ machine_code_byte
190 POKE address,machine_code_byte
200 LET address=address+1
210 END REPeat read_and_store_machine_code
220 CALL reserved_address
230 PRINT #0,"Now try pressing break..."
240 REPeat forever
250 FOR border_width=99 TO 0 STEP -4
260 BORDER border_width,RND(0 TO 7)
270 END FOR border_width
280 END REPeat forever
290 :
300 DATA 67,250,0,16,65,250,0,46,33,73
310 DATA 0,4,112,30,78,65,78,117,32,121
320 DATA 0,2,128,16,67,232,0,104,74,233
330 DATA 0,143,74,104,0,20,108,6,49,124
340 DATA 255,254,0,20,66,57,0,2,128,51
350 DATA 78,117,0,0,0,0,0,0,0,0
```



LISTADO 4

```
PC_TCTRL          EQU       $1B002
PC_MCTRL          EQU       $1B020
SV_TMODE          EQU       $2B040
MD_SECTR          EQU       $012A
MOTORS_OFF_AH     EQU       $2B5A               DIRECCION EN 'AH' (VER-
                                                SION 1.02)
MOTORS_OFF_JM     EQU       $2B64               EN 'JM' (V. 1.03)
MOTORS_OFF_JS     EQU       $2C50               EN 'JS' (V. 1.10)
MOTORS_OFF_JSU    EQU       $2C9A               EN 'JSU'(V. 1U10)
MOTORS_OFF_MG     EQU       $2CCC               EN 'MG' (V. 1.13)
MOTOR_ON_AH       EQU       $2B64               DIRECCION EN 'AH' DE LA
                                                RUTINA DE MOTOR ENCENDI-
                                                DO DEL MICRDRIVE
MOTOR_ON_JM       EQU       $2B70               EN 'JM'
MOTOR_ON_JS       EQU       $2C56               EN 'JS'
MOTOR_ON_JSU      EQU       $2C9A               EN 'JSU'
MOTOR_ON_MG       EQU       $2CD2               EN 'MG'
TIME_OUT          EQU       1000
PC_NOTMD          EQU       %11100111
PC_SERB           EQU       %00010000
INTS_OFF          EQU       %0000011100000000
INTS_ON_USER      EQU       %1101100011111111
ERR_OK            EQU       0
ERR_NF            EQU       -7
ERR_EF            EQU       -16

* COMIENZO DE LA RUTINA DE LECTURA DE CABECERA DE UN  SECTOR  DE
  CARTUCHO MICRODRIVE

                  TRAP      ^0
                  CMP.L     ^9,D1
                  BGE.S     NOT_FOUND
                  TST.L     D1
                  BEQ.S     NOT_FOUND
                  ORI.W     ^INTS_OFF,SR
                  LEA       PC_MCTRL,A3
                  MOVE.B    ^PC_NOTMD,DO
                  AND.B     SV_TMODE,DO
                  ORI.B     ^PC_SERB,DO
                  MOVE.B    DO,PC_TCTRL
                  MOVE.B    DO,SV_TMODE
                  MOVEA.L   ^MOTOR_ON_JS,AO
                  JSR       (AO)
                  MOVE.W    ^TIME_OUT,D3
BAD_HEADER        SUBQ.W    ^1,D3
                  BEQ.S     BAD_MEDIUM
                  LEA       HEADER_BUFFER,A1
                  MOVEA.W   MD_SECTR,AO
                  JSR       $4000(AO)
                  BRA.S     BAD_MEDIUM
                  BRA.S     BAD_HEADER
                  MOVEQ     ^ERR_OK,D7
                  BRA.S     FINISH
NOT_FOUND         MOVEQ     ^ERR_NF,D7
                  BRA.S     QUIT
BAD_MEDIUM        MOVEQ     ^ERR_EF,D7
FINISH            MOVEA.L   ^MOTORS_OFF_JS,AO
                  JSR       (AO)
                  MOVE.B    ^PC_NOTMD,DO
                  AND.B     SV_TMODE,DO
                  MOVE.B    DO,PC_TCTRL
                  MOVE.B    DO,SV_TMODE
QUIT              ANDI.W    ^INTS_ON_USER,SR
                  MOVE.L    D7,DO
                  RTS

* ALMACENAMIENTO PARA CABECERA DE SECTOR

HEADER_BUFFER
FLAG              DS.B      1
SECTOR_NUMBER     DS.B      1
NAME              DS.B      10
RAND_NUMBER       DS.W      1
                  END
```

El método sólo es efectivo con programas SuperBASIC sencillos, aunque el teclado puede seguir leyéndose empleando KEYROW (por ejemplo, para leer el cursor o las teclas de función).

UN METODO MAS EVOLUCIONADO PARA LA DESACTIVACION DE BREAK

La desactivación de BREAK puede realizarse de una manera más efectiva con el empleo de una rutina adicional de planificación *(Scheduler Loop Task).* El propio sistema operativo del QL posee una rutina de este tipo, consistente en un programa que controla y planifica la secuencia de tareas y la distribución de los medios de la máquina. En nuestro caso, será solicitada cada vez que se produzca un llamamiento a la rutina de mantenimieto del QDOS, ejecutándose siempre antes que esta última.

Para que pueda desactivarse BREAK, primero tenemos que asegurarnos que la variable del sistema BV_RBK en la dirección $008F(A6) nunca contenga 0 y que la tarea Super-BASIC siempre esté activa. También debe desactivarse el efecto de CTRL + F5 para asegurar que siempre esté en funcionamiento la pantalla, estableciendo adecuadamente el contenido de la variable del sistema asociada con ella.

El programa ensamblador del listado 2 establece una tarea de planificación para que BV_BRK nunca contenga 0 (con lo que se evita que el QDOS advierta la pulsación de BREAK) e inspecciona el estado de la tarea SuperBASIC alterándolo en consecuencia. La rutina entra siempre en funcionamiento antes que las tareas de planificación de la ROM, las cuales incluyen la secuencia de activación de BREAK.

La instrucción TAS *(Test and Set)* usada en el programa, analiza y establece el *bit* 7 del *byte* de operando en una operación indivisible. Esta es una forma limpia de señalar que BREAK no se ha pulsado, en lugar de cargar $80 en la



variable del sistema correspondiente.

Esta rutina puede incorporarse en el interior de su archivo BOOT utilizando el método descrito anteriormente. En caso de no disponer un ensamblador, utilice el programa del listado 3, que contiene un cargador de código máquina y una demostración de sus efectos.

COMO IMPEDIR LA COPIA DEL CARTUCHO ORIGINAL

Aparte de la utilización de cartuchos ROM, el mejor medio de protección en el QL consiste en obligar al comprador de *software* a mantener su cartucho original.

Para llevar a cabo este método, debe introducirse el cartucho en el *microdrive* 2 y leer una cabecera de sector, obteniéndose un número aleatorio de 16 *bits,* creado en el momento del formateo. Si el número registrado en el cartucho original se lee durante su creación y se carga en una tabla en el cargador del programa que necesita protección, este cargador puede compararlo con el número leído en el cartucho del *microdrive* 2. En caso de no ser el mismo, o no existir cartucho, entrará en funcionamiento un sistema de reinicialización. En caso contrario, el cargador seguirá cargando el programa principal.

Por supuesto, el pirata tiene todavía una posibilidad entre 65536 de formatear su propio cartucho con el mismo número aleatorio de 16 *bits* que el del original. Sin embargo, esto bastará para obligarle a desistir de sus propósitos.

Cargando el programa del listado 5, podrá hacer funcionar el código máquina, que colocará un número aleatorio en el cartucho del *microdrive* 1. A continuación, deberá introducir otro cartucho (o el mismo) en el *microdrive* 2. Entonces, el programa lleva a cabo la comparación y le dirá si es o no es el mismo cartucho.

Las rutinas ROM para poner en marcha y parar los motores de los *microdrives* no son vectoriales como las de lecturas de cabecera de sectores, sin embargo, en futuras versiones del QL se podrá remediar esto. De momento, debe conocer cuál es su versión de QL para poder modificar el programa.

Pasa saber cuál es su versión de ROM, deberá utilizar la función VER$. El programa del listado 4 ha de ensamblarse incluyendo las direcciones correctas que necesita su versión. En cuanto al programa de demostración SuperBASIC (listado 5) las direcciones necesarias se POKEan de forma automática.

Orlando Araujo Martín

LISTADO 5

```
100 MODE 4
110 LET reserved_address=RESPR (144)
120 LET address=reserved_address
130 RESTORE
140 REPeat read_and_store_machine_code
150 IF EOF THEN EXIT read_and_store_machine_code
160 READ machine_code_byte
170 POKE address,machine_code_byte
180 LET address=address+1
190 END REPeat read_and_store_machine_code
200 LET rom_ver$=VER$
210 IF rom_ver$(1 TO 2)='PM' OR rom_ver$='FB' THEN
220 PRINT "Return QL to Sinclair"
230 STOP
240 END IF
250 IF rom_ver$ (1 TO 2)='AH' THEN
260 POKE_W reserved_address+54,11108
270 POKE_W reserved_address+96,11114
280 END  IF
290 IF rom_ver$(1 TO 2)='JM' THEN
300 POKE_W reserved_address+54,11120
310 POKE_W reserved_address+96,11114
320 END IF
330 IF rom_ver$(1 TO 3)='JSU' THEN
340 POKE_W reserved_address+54,11424
350 POKE_W reserved_address+96,11418
360 END IF
370 IF rom_ver$(1 TO 2)='MG' THEN
380 POKE_W reserved_address+54,11474
390 POKE_W reserved_address+96,11468
400 END IF
410 REPeat test_microdrive_cartridges
420 PRINT "Place a cartridge in mdv1 and press a key"
430 PAUSE
440 CALL reserved_address,1
450 LET mdv_one_number=PEEK_W (reserved_address+132)
460 PRINT "place a cartridge in mdv2 and press a key"
470 PAUSE
480 CALL reserved_address,2
490 LET mdv_two_number=PEEK_W (reserved_address+132)
500 IF mdv_one_number=mdv_two_number THEN
510 PRINT "Cartridge in mdv2 same as that in mdv1"
520 ELSE
530 PRINT "cartridge in mdv2 not the same as in mdv1"
540 END IF
550 END REPeat test_microdrive_cartridges
560 :
570 DATA 78,64,178,188,0,0,0,9,108,76
580 DATA 74,129,103,72,0,124,7,0,71,249
590 DATA 0,1,128,32,16,60,0,231,192,57
600 DATA 0,2,128,168,0,0,0,16,19,192
610 DATA 0,1,128,2,19,192,0,2,128,160
620 DATA 32,124,0,0,44,86,78,144,54,60
630 DATA 3,232,83,67,103,24,67,250,0,62
640 DATA 48,120,1,42,78,168,64,0,96,10
650 DATA 96,236,126,0,96,6,126,249,96,32
660 DATA 126,240,32,124,0,0,44,80,78,144
670 DATA 16,60,0,231,192,57,0,2,128,160
680 DATA 19,192,0,1,128,2,19,192,0,2
690 DATA 128,160,2,124,216,255,32,7,78,117
700 DATA 0,0,0,0,0,0,0,0,0,0
710 DATA 0,0,0,0
```



ICE: Iconos, ventanas y ratones

Hasta hace poco, el usuario sin conocimientos de informática debía entablar una lucha desigual con su ordenador. Los sistemas operativos obligaban al aprendizaje de enormes listas de comandos y sintaxis. El Lidsa de Apple, mediante el empleo de un interface de usuario basado en ventanas, iconos, ratón y menus pull down, inició un proceso de simplificación en la utilización de ordenadores personales. Al Lisa le sucedió el Macintosh, con características similares y un precio más atractivo. Con el ICE (Icon Control Environment), el QL se incorpora a la misma tendencia.

Cuando se conecta un ordenador provisto de estas facilidades, aparece en pantalla la superficie de un escritorio con diversos objetos situados sobre él: son los iconos. Cada uno de ellos posee una función determinada, que se corresponde con comandos del sistema operativo. Para facilitar la selección se hace uso de unratón, dispositivo que al ser desplazado sobre la mesa provoca el movimiento de un cursor en la pantalla. Colocado dicho cursor sobre el icono seleccionado, se pulsa una tecla incorporada al ratón y el ordenador ejecuta la tarea solicitada. Al igual que en un escritorio verdadero, podemos superponer en la pantalla diversas ventanas a modo de documentos con la información que necesitamos consultar. De esta manera, el uso del teclado queda reducido al mínimo.

El Macintosh, de Apple Computer, incorpora un sistema de iconos plenamente evolucionado, y su facilidad de manejo ha marcado un hito en la historia de los ordenadores personales. Recientemente, Digital Research ha desarrollado GEM, un entorno de iconos, ventanas y ratón del que existen versiones para diversas máquinas, como el IBM PC, el Atari 520 ST, y otras.

La idea de facilitar el trabajo y aprendizaje del usuario es sin duda la última moda. Sin embargo, comenzó a forjarse a comienzos de la década de los 70, en el centro de Xeros Corp en Palo Alto, California, cuando los investigadores de esta empresa decidieron que eran los ordenadores los que habían de adaptarse a las personas, y no al contrario, como venía sucediendo.

ICONOS EN EL QL

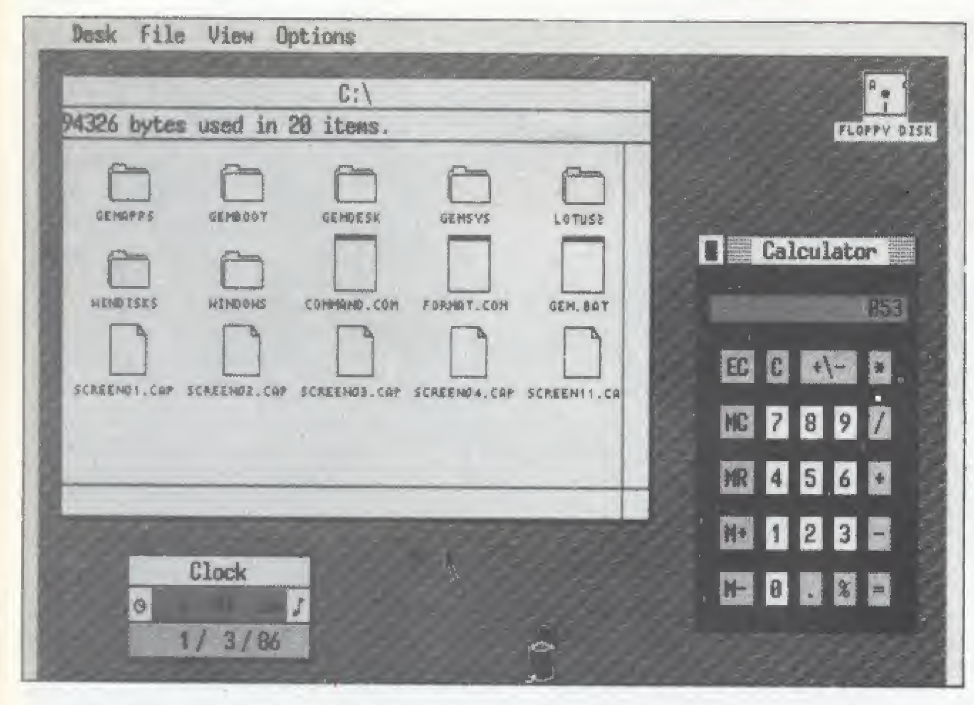
El QL también se ha querido incorporar a estas innovaciones. Basándose en los sistemas ya existentes, la compañía Eidersoft ha creado el Icon Control Environment (ICE), suministrado en un cartucho ROM. Diseñado como «herramienta amable» de cara al usuario para hacer frente al complejo QDOS, el ICE tiene mucho en común con los sistemas de iconos y ventanas de otros ordenadores. Esto permite proteger al usuario de los desvaríos del sistema operativo gracias a un extenso surtido protector a base de gráficos. La idea fundamental del ICE consiste en aproximar el manejo del ordenador a la gestión habitual de una oficina.

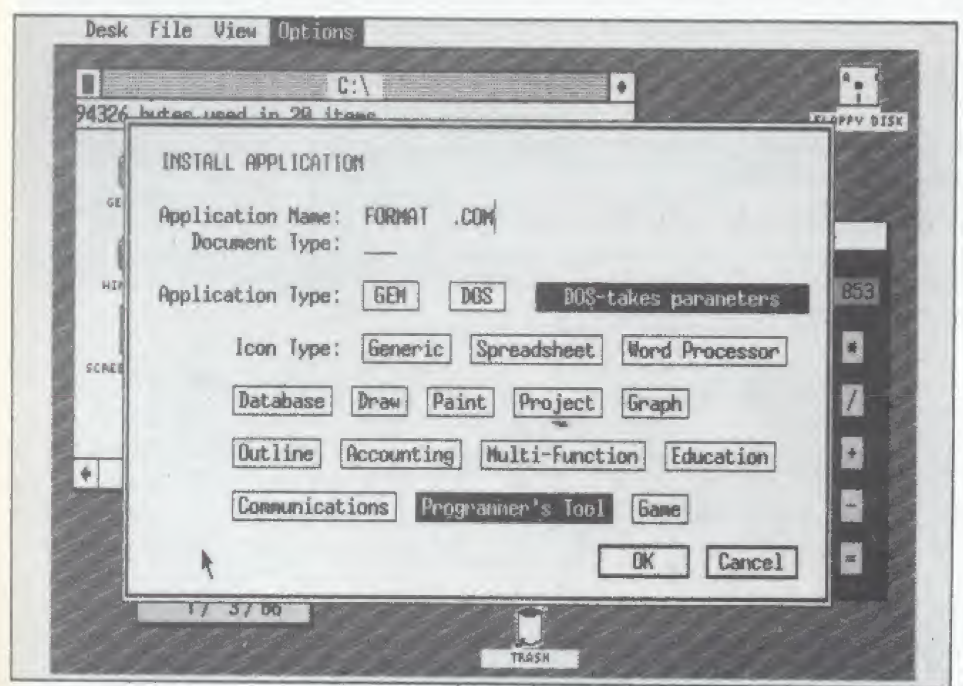
Después de enchufar el QL con la EPROM ICE de 16 K insertada en el conector de ampliación de memoria ROM, en la pantalla aparece un escritorio sobre el que hay dibujadas diversas figuras o iconos que identifican las diferentes tareas que puede realizar el QL, archivos disponibles y varias facilidades como calendario, calculadora y papelera.

COMODIDAD Y RAPIDEZ DE MANEJO

Para utilizar el ICE, todo lo que el usuario debe hacer es limitarse a desplazar por la pantalla una flechita a modo de cursor o puntero mediante las teclas de desplazamioento, joystick o ratón (este último estará disponible próximamente en nuestro país). Cuando el puntero está junto al icono elegido por el usuario o cerca de una de las opciones de control (situadas en la parte inferior de la pantalla) lo único que hay que hacer es pulsar la barra de espaciado una o dos veces, dependiendo de si desea selec-







GEM en IBM PC. Su parecido con el sistema del Macintosh no puede ser mayor.

cionar la tarea o inicializar cualquier operación implicada. El proceso, que no requiere ninguna introducción de datos desde el teclado, es prácticamente un juego de niños.

Las operaciones básicas de mantenimiento, como pueden ser la extracción de directorios, copia de archivos y ejecución de programas, se llevan a cabo de forma implícita por el ICE. El usuario sólo se limita a señalar y pulsar, ICE hace el resto. Por ejemplo, si se ve en la necesidad de copiar o borrar varios archivos, basta señalar el archivo en cuestión y llevarlo después a su dispositivo de destino o a la papelera según tenga que copiar o destruir el archivo señalado. Si las operaciones a realizar son de carácter irreversible, ICE pedirá al usuario la confirmación de sus intenciones. Cuando se producen errores, la pantalla muestra los mensajes correspondientes, escritos, por supuesto en inglés (aunque bastante sencillos de comprender).

GESTION DE ARCHIVOS

Los archivos en disco y *microdrive* se paginan en la pantalla en grupos de 12. ICE es capaz de distinguir automáticamente diferentes clases de archivos, utilizando diferentes iconos para representarlos. Las clases de ficheros que puede reconocer son archivos generales en código máquina, programas en código máquina ejecutables (EXEC), programas en SuperBASIC, documentos QUILL, hojas del ABACUS, gráficos del EASEL, y bases de datos del ARCHIVE, así como archivos de ejecución automática (boot). La mayor ventaja que se desprende en este caso, es que el usuario puede saber mediante un simple vistazo el contenido de un archivo. Otro punto menos evidente consiste en que para hacer funcionar un programa ya no es necesario distinguir entre archivos que se ejecutan con EXEC o RUN. Los programas cargadores de los paquetes de PSION (versión 2.000) ya no son necesarios.

El mantenimiento general y las operaciones del sistema aparecen de modo explícito como opciones en un panel de control en la parte inferior de la pantalla. Estos permiten cambiar el nombre de archivo (RENAME), inspeccionar sus contenidos (VIEW) y solicitar información acerca de uno de ellos (INFO), con lo cual conoceremos el nombre completo del archivo, longitud y tipo y últimos datos rectificados o añadidos. Con las opciones de este panel también se pueden formatear diskettes o cartuchos de *microdrive (FORMAT)* y solicitar información referente a la memoria disponible, rutinas que están funcionando y la versión del QDOS o ICE que se está utilizando. La opción CUSTOM permite variar según las necesidades del usuario, los datos (que normalmente funcionan por omisión con unos valores predeterminados) referentes a la fecha, hora, nombres de dispositivos, instalación de impresora, velocidad del cursor y sensibilidad de las teclas. Finalmente, existe una opción para



MacCoach, un programa para aprender a manejar el Macintosh.

abandonar el ICE y volver al SuperBASIC.

FUNCIONES SUPLEMENTARIAS

ICE incorpora dos funciones especiales: calendario y calculadora. Utilizando la primera, la pantalla muestra un calendario completo con todos los meses, marcando la fecha actual iluminándola en rojo. Con el empleo de la segunda, aparece en la pantalla la parte frontal de una típica calculadora de bolsillo, con sus teclas y pantalla de visualización de datos. Incluye una tecla de memorización y porcentaje y funciona desplazando el cursor sobre las teclas.

Al adquirir la EPROM ICE, Eidersoft también proporciona al comprador de un cartucho de *microdrive* que contiene una serie de programas (para utilizarlos conjuntamente con el ICE) que le permiten hacer funcionar en multitarea cuatro programas al mismo tiempo. Este paquete de programas (llamado CHOICE) puede ser de gran utilidad con una ampliación de 256 *Kbytes* de memoria, ya que si reservamos 128 *Kbytes* para ARCHIVE, ABACUS y QUILL (por ejemplo) podremos hacerlos funcionar simultáneamente y acceder individualmente a cada uno de ellos pulsando simplemente CRTR + F3. No solamente es impresionante la velocidad con la que nos movemos entre las distintas aplicaciones, sino también el hecho de que cuando volvemos a una de ellas, nos encontramos exactamente en el mismo punto donde las dejamos antes, ya sea documento, base de datos u hoja de cálculo.



Pantalla de QL con ICE. Se observa la calculadora y varios iconos.

Al no tener un elaborado sistema de ventanas, el ICE es mucho menos sofisticado que otros programas semejantes. Sin embargo, hay que tener en cuenta que el programa solamente ocupa 16 K, en lugar de los 90 K de su hermano mayor GEM. De todos modos, es fácil de utilizar y está diseñado a prueba de errores. En combinación con el paquete CHOICE, abre una nueva dimensión para el usuario que no había logrado sacar el máximo fruto de su QL hasta este momento.

Nota: ICE se comercializa en el Reino Unido a un precio aproximado de 60 libras.

Orlando Araujo Martín

CATALOGO DE SOFTWARE PARA ORDENADORES PERSONALES IBM

TODO EL CATALOGO DE SOFTWARE CON MAS DE 800 FICHAS





1.ª ENTREGA 3.500,— PTAS.
(400 FICHAS + FICHERO)

RESTO EN TRES ENTREGAS TRIMESTRALES DE 1.500,— PTAS. CADA UNA.

PRECIO TOTAL DE LA SUSCRIPCION - 8.000,— PTAS.

CUPON DE PEDIDO

SOLICITE **HOY MISMO** EL CATALOGO DIRECTAMENTE A

infodis, s.a.

BRAVO MURILLO, 377 - 5.° A
28020 MADRID

O EN LOS CONCESIONARIOS IBM

El importe lo abonaré: POR CHEQUE ☐ CONTRA REEMBOLSO ☐
CON MI TARJETA DE CREDITO ☐ Ref: CATALOGO DE SOFTWARE

Cargue **8.000** ptas. a mi tarjeta American Express ☐ Visa ☐ Interbank

Número de mi tarjeta ______

Fecha de caducidad ______ Firma

NOMBRE ______

CALLE ______

CIUDAD ______ D.P. ______

PROVINCIA ______

NUEVOS COMANDOS

DEL SPECTRUM 128 K

De las 32 K de ROM del 128 (en realidad EPROM), 16 son básicamente una copia de la del Plus (o del 48 K o del 16 K), mientras que las otras 16 implementan un nuevo editor, el sistema de paginación, los mensajes de error en castellano y algunos comandos adicionales que podemos disfrutar desde tres modos de edición diferentes.

Cuando ya muchos han adquirido la última novedad de Sinclair-Investrónica y parece causar entre sus usuarios, cualquier reacción excepto la indiferencia, es hora de sentarse a observar detenidamente lo que nos ofrece de positivo su sistema operativo respecto al del Spectrum Plus.

Cuando se inicializa el ordenador, aparece el *copyright* de Sinclair y la palabra «ESPAÑOL». Si pulsamos cualquier tecla en el modo COMANDO, con el cursor en la parte baja de la pantalla y sobre una línea en INVERSE que nos indica si estamos en modo MAYUSCULAS, GRAFICO o EXTENDIDO. Podremos utilizar el teclado numérico (y también el principal) como si de una calculadora se tratase, es decir, tecleando directamente 5-3 obtendremos como resultado 2, y si continuamos con *7 aparecerá un hermoso 14 en pantalla (lo de «hermoso» es una forma de hablar, el juego de caracteres es idéntico al del 48 K).

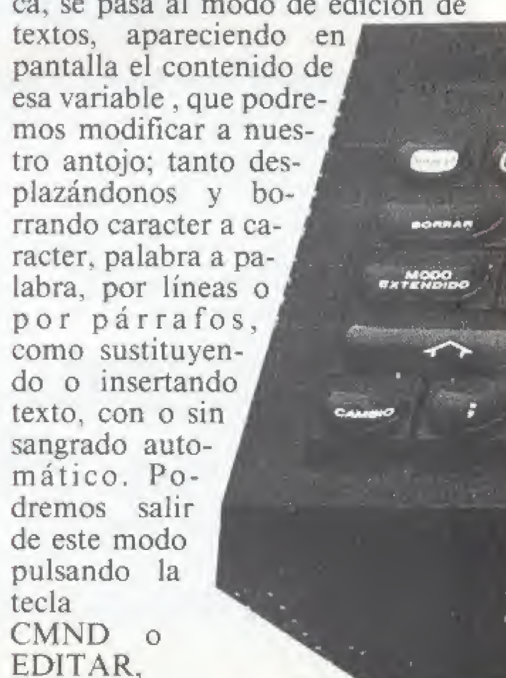
Podremos también desde este modo disfrutar de todos los nuevos comandos tecleándolos directamente, letra a letra y en mayúsculas (para mayor comodidad algunos podrán ser reducidos hasta el límite de que bastará con teclear la inicial). Podremos encadenar varios comandos (con «:»), pero, en este modo, la línea resultante en ningún caso podrá superar los 32 caracteres.

Disponemos de la posibilidad de renumerar las líneas BASIC mediante el comando NUMERO (o N), que debe ir seguido de dos valores separados por una coma: el primero es el que le daremos a la primera línea, y el segundo es el «paso» entre líneas. Esto renumerará sin problemas los GOTOs y GOSUBs normales, pero ignorará los que contengan variables u operaciones aritméticas.

El comando BORRAR (o B) resultará útil si queremos eliminar bloques BASIC. Deben seguirle dos parámetros, la primera y la última línea a borrar, separados por una coma.

Pseudoprocesador de textos

El comando EDIT (o E) puede actuar de dos formas. Si le sigue un espacio y una variable alfanumérica, se pasa al modo de edición de textos, apareciendo en pantalla el contenido de esa variable , que podremos modificar a nuestro antojo; tanto desplazándonos y borrando caracter a caracter, palabra a palabra, por líneas o por párrafos, como sustituyendo o insertando texto, con o sin sangrado automático. Podremos salir de este modo pulsando la tecla CMND o EDITAR,



SPECTRUM
128

INVESTRONICA

BMC

POWER

V-HOLD H-POSITION BRIGHT

PARAR

CAMBIO

128K

e imprimir el texto o salvarlo en cinta como una matriz de datos. Por ejemplo, si estábamos escribiendo en a$ bastará LPRINT a$ o SAVE "nombre" a$(). Si tenemos un texto en alguna variable, hay que poner cuidado en salvarla en cinta o disco RAM (más adelante explicaremos lo que es esto) antes de hacer CLEAR, RUN o crear una variable con el mismo nombre, pues sino perderíamos todo lo hecho.

Un editor BASIC más versátil

Si el comando EDIT va seguido de un número, se pasa el modo de acción de líneas BASIC, con el cursor situado en el comienzo de la línea especificada (por omisión la primera) o la inmediata superior. También se entra en este modo pulsando las teclas ENTER y EDITAR. Podremos desplazarnos por las líneas en horizontal, vertical, caracter a caracter o línea a línea, borrando e insertando lo que que-

Una de las mayores virtudes del 128 es que nos permite por medio del comando SPECTRUM hacerle olvidar que tiene RAM y ROM extra y que se comporte como un Plus

remos. Cuando pasamos de una línea a otra se controla la sintaxis de la que abandonamos y se nos avisa, si hay algún error, con un simpático *bug*.

Pulsando ENTER en este modo, el editor abrirá un hueco entre las líneas, donde podremos introducir comandos directos (no incluidos NUMERO, BORRAR o EDIT). Cabe destacar que si tecleamos varios comandos seguidos y separados por «:», automáticamente comenzará cada uno de ellos una nueva línea, con lo que el resultado es una mayor claridad en los listados. Asimismo, aunque tecleemos los comandos sin dejar espacios entre sus componentes (p. e., 10FORNN=0TO9STEP3), aparecerán en los listados perfectamente presentados (es decir: 10 FOR N=0 TO 9 STEP 3).

Es posible abandonar este modo y pasar a la parte baja pulsando las teclas CMND o EDITAR, pero si lo hacemos después de haber retocado una línea y antes de que ésta haya sido reconocida (para lo cual habremos de salir de ella), las modificaciones realizadas no serán tomadas en cuenta.

Comando SPECTRUM, disco RAM y RS232

Una de las mayores virtudes del 128 es que nos permite, por medio del comando SPECTRUM, hacerle olvidar que tiene RAM y ROM extra y que se comporte como un



Plus (salvo algún detalle, como los mensajes de error que seguirán en castellano). Es esta propiedad la que lo haces compatible con la mayor parte del *software* disponible para Plus (o 48 K o 16 K).

Debido a lo molesto de los sistemas de paginación de memoria, el BASIC del 128 seguirá disponiendo sólo de 48 K RAM, las 80 K extras podrán ser utilizadas para correr programas en código máquina o desde el BASIC como disco RAM. Este es un sistema de almacenamiento de datos ultra-rápido, pero con un grave inconveniente: los datos se pierden en cuanto se desconecta la alimentación. Aun así, resulta muy útil para determinadas aplicaciones. Los comandos que acceden a él son iguales a los del cassette pero seguidos por el signo «i». Serán válidos por tanto:

SAVEi "nombre" LINE 10
LOADi "nombre" CODE 50000,100
SAVEi "nombre" SCREEN$
MERGEi "nombre" DATA a$()

El comando VERIFY no es en realidad necesario, pues este tipo de almacenamiento es tan fiable como el posible proceso de verificación, pero hemos comprobado que lo admite, aunque no realiza su cometido e incluso puede llegar a colgar el sistema.

Para hacer un listado de los ficheros contenidos en el disco RAM podremos utilizar CATi, y para borrar cualquiera de ellos servirá ERASEi "nombre".

Podremos conectar el ordenador con cualquier impresora compatible RS232. Para ello existe el comando FORMAT «P»; que debe ir seguido de la velocidad en baudios con que opera la impresora. El canal 3 queda inicialmente abierto al acceso RS232, por lo que toda la salida que le afecte lo hará por éste.

Debido a lo molesto de los sistemas de paginación de memoria, el BASIC del 128 dispone sólo de 48 K. Las 80 K restantes pueden utilizarse desde código máquina

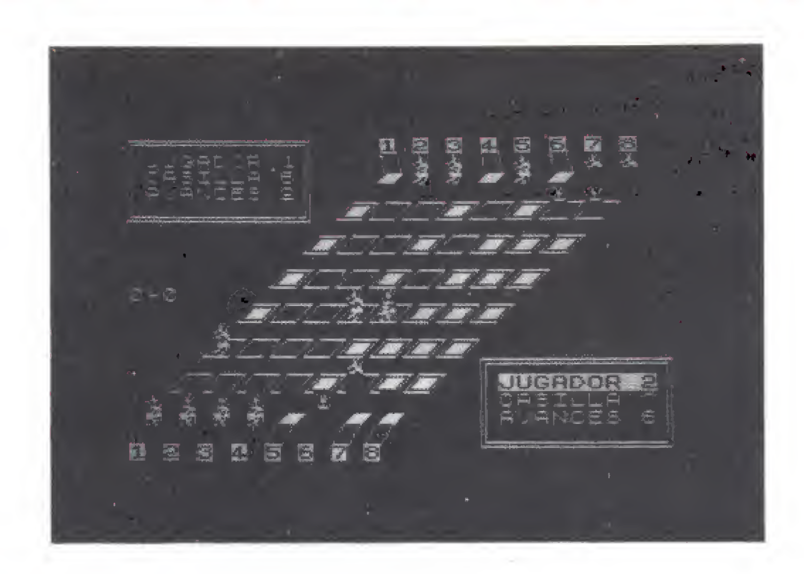


PLAY o la forma de hacer mucho ruido

Sin duda el comando más completo de los que nos ofrece de novedad el 128 es PLAY. Tiene que ir seguido de una a tres cadenas o variables alfanuméricas separadas por comas, cada una de las cuales corresponde a uno, dos o tres de los seis canales de sonido disponibles. En ellas podremos especificar frecuencias, octavas, duraciones, ligaduras, volúmenes, envolventes, velocidades de ejecución, repeticiones, comentarios, etc. Aunque resulte algo complicado controlar todas estas opciones a quien, de por sí, entienda poco de música, este comando permitirá que los programadores creativos dispongan de un buen repertorio de melodías y efectos sonoros.

Otro de los puntos fuertes de este comando es que permite controlar, por medio de la misma, salidas que el RS232, cualquier instrumento electrónico que posea MIDI *(Musical Instruments Digital Interface)*. La mayoría de los sintetizadores y cajas de ritmo lo tienen, por lo que podremos hacer que interpreten melodías escritas en el ordenador.

En resumen, aunque se podría haber depurado algo más el diseño de este «hijo español» de Sinclair, las modificaciones y añadidos se agradecen, y llenan algo el hueco que dejaba el Spectrum y explotaban los MSX, Commodore o Amstrad. Esperemos que **Investrónica** aproveche que el nuevo sistema operativo reside en EPROM para, sin perder la compatibilidad, limar los pequeños *bugs* e ir mejorando esta ampliación del operativo más amado, odiado y destripado de la historia de los ordenadores domésticos.



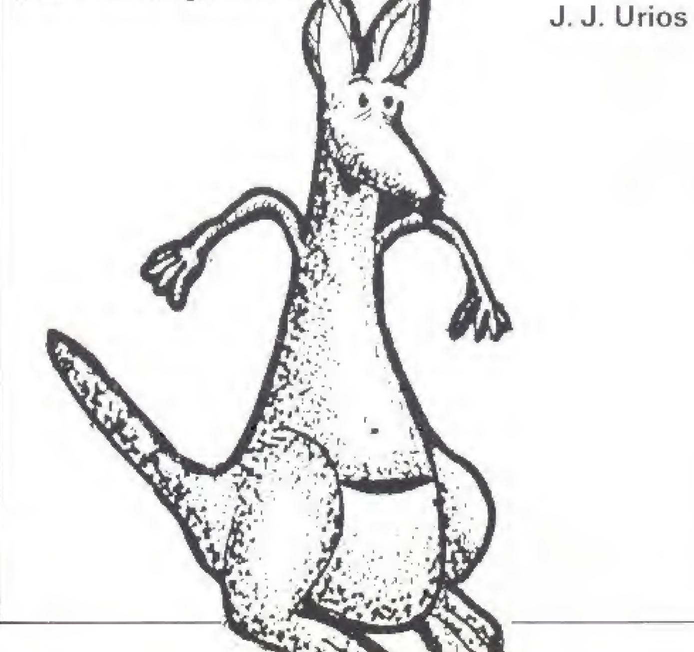
Reunión K

Sin otro propósito que el pasar un buen rato intentando derrotar al adversario (humano o Spectrum), Reunión K es un juego de lógica y estrategia con una magnífica presentación. Entre sus características figuran:

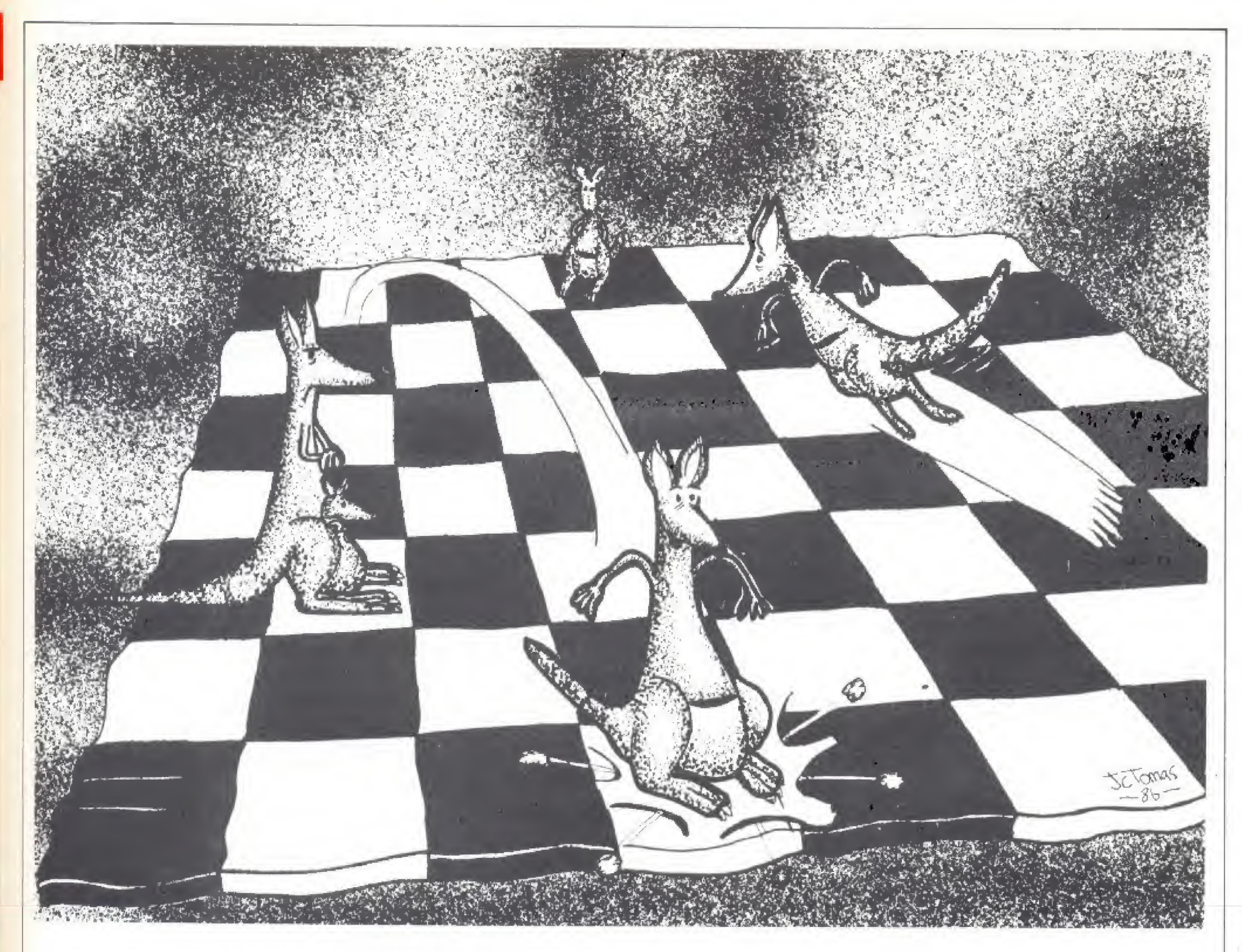
- Gráficos tridimensionales
- Sonido
- Tres niveles de dificultad
- Marcador multipartidas
- Instrucciones incorporadas

El jugador que consiga el último encuentro entre canguros (el nombre del juego significa precisamente reunión de Kanguros) se proclamará ganador.

J. J. Urios

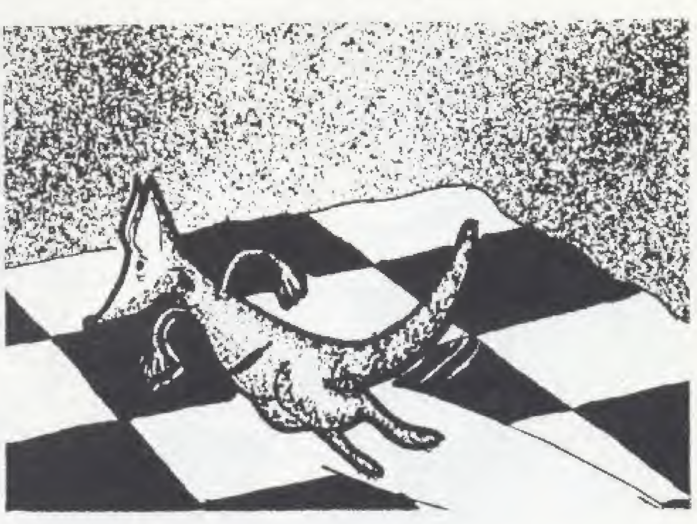


```
  10 LET tju1=0: LET tju2=0: GO
SUB 5100
  14 INK 6: PAPER 0: BORDER 0: C
LS
  15 GO SUB 9000
  20 INK 6: CLS
  22 LET ronda=0
  25 PRINT AT 6,8;"ELIGE OPCION"
  30 PRINT AT 12,2;"2.  DOS JUGA
DORES"
  32 PRINT AT 13,2;"3.  JUEGAS C
ONTRA EL ORDENADOR"
  35 IF INKEY$="" THEN  GO TO 35
  36 BEEP .02,68
  37 IF INKEY$="3" THEN  LET nju
=3: GO TO 40
  38 IF INKEY$="2" THEN  LET nju
=2: GO TO 40
  39 GO TO 35
  40 LET tur=1
  60 IF INKEY$<>"" THEN  GO TO 6
0
  61 IF nju=2 THEN  GO TO 99
  62 CLS
  65 PRINT AT 6,8;"ELIGE OPCION"
  66 PRINT AT 12,2;"  NIVEL DE D
IFICULTAD (1-3)"
  72 IF INKEY$="" THEN  GO TO 72
  74 BEEP .02,68
  78 IF INKEY$>"0" AND INKEY$<"4
" THEN  LET dif=VAL INKEY$: GO T
O 98
  80 GO TO 65
```




```
  98 IF INKEY$<>"" THEN  GO TO 9
8
  99 CLS : GO SUB 5000: PRINT AT
 9,0;tju1;"-";tju2
 100 IF nju=2 AND tur=1 THEN  GO
 SUB 1000: GO SUB 2000: GO TO 10
0
 110 IF nju=2 AND tur=2 THEN  GO
 SUB 2000: GO SUB 1000: GO TO 10
0
 120 IF nju=3 AND tur=1 THEN  GO
 SUB 1000: GO SUB 3000: GO TO 10
0
 130 GO SUB 3000: GO SUB 1000: G
O TO 100
1000 LET opc=1: LET ord=0: GO SU
B 4000: RETURN
2000 LET opc=2: GO SUB 4000: RET
URN
3000 REM tirada de ordenador
3001 LET ronda=ronda+1
3005 LET opc=2: LET ord=1
3010 DIM g(3): DIM r(8)
3015 LET eq=0: LET alt=0
3020 FOR n=1 TO 8
3026 IF alt=t(n,3) THEN  LET eq=
eq+1: LET r(eq)=n
3027 IF alt<t(n,3) THEN  LET alt
=t(n,3): LET eq=1: LET r(eq)=n
3030 LET x=t(n,3): IF x>3 THEN
LET g(3)=g(3)+1: LET x=x-4
3040 IF x>1 THEN  LET g(2)=g(2)+
1: LET x=x-2
3050 IF x>0 THEN  LET g(1)=g(1)+
1
3060 NEXT n
3070 LET y=0
3080 FOR n=1 TO 3: IF g(n)<>2*IN
T (g(n)/2) THEN  LET y=y+n+(n=3)
3085 NEXT n
3090 IF y<>0 AND ronda>10-dif*3
THEN  GO TO 3120
3100 LET ca=r(1+INT,(RND*eq)): L
ET av=1+INT (t(ca,3)*RND): GO SU
B 4000: GO TO 3999
3120 FOR n=1 TO 8: LET r(n)=n: N
EXT n
3125 FOR n=1 TO 8
3130 LET rnd=1+INT (RND*(9-n))
3135 LET ca=r(rnd)
```

```
3140 LET r(rnd)=r(9-n)
3142 IF t(ca,3)=0 THEN  GO TO 32
20
3145 LET guar=t(ca,3)
3150 FOR m=1 TO t(ca,3)
3170 LET t(ca,3)=t(ca,3)-m
3175 LET y=0
3180 GO SUB 3500
3190 LET t(ca,3)=guar
3195 BEEP .003,17: PRINT AT 11,2
6; INK m+1;"PIENSO"
3200 IF y=0 THEN  LET av=m: GO T
O 3230
3210 NEXT m
3220 NEXT n
3230 PRINT AT 11,26;"      "
3240 GO SUB 4000
3499 RETURN
3500 DIM g(3): FOR k=1 TO 8
3510 LET x=t(k,3): IF x>3 THEN
LET g(3)=g(3)+1: LET x=x-4
3520 IF x>1 THEN  LET g(2)=g(2)+
1: LET x=x-2
3550 IF x>0 THEN  LET g(1)=g(1)+
1
3560 NEXT k: LET y=0
3580 FOR k=1 TO 3: IF g(k)<>2*IN
T (g(k)/2) THEN  LET y=y+k+(k=3)
3590 NEXT k
3999 RETURN
4000 PRINT  INK opc+4;AT (opc=1)
+(opc=2)*14,(opc=1)+(opc=2)*22;
INVERSE 1; FLASH 1;"JUGADOR ";op
c
4010 PRINT AT (opc=1)*2+(opc=2)*
15,(opc=1)+(opc=2)*22; INK opc+4
;"CASILLA "; FLASH 1;"?";CHR$ 8;
4015 LET clo=0
4020 LET m$=INKEY$: IF ord=1 THE
N  LET m$=STR$ ca
4022 LET clo=clo+1: IF clo=30 TH
EN  LET clo=0: BEEP .02,RND*-20
4024 IF m$="" THEN  GO TO 4020
4030 BEEP .002,44
4040 IF m$<"1" OR m$>"8" THEN  G
O TO 4020
4050 IF t(VAL m$,3)=0 THEN  PRIN
T  INK opc+4;AT 21,0;"CASILLA ";
m$;" BLOQUEDA. ELIGE OTRA": BEEP
 .5,0: BEEP .09,-10: BEEP .3,-2:
 GO TO 4000
4060 LET ca=VAL m$: PRINT m$
4070 PRINT AT 21,0;"
                               "
4075 IF INKEY$<>"" THEN  GO TO 4
075
```



```
4080 PRINT AT (opc=1)*3+(opc=2)*
16,(opc=1)+(opc=2)*22; INK opc+4
;"AVANCES "; FLASH 1;"?";CHR$ 8;
4090 LET m$=INKEY$: IF ord=1 THE
N  LET m$=STR$ av
4095 LET clo=clo+1: IF clo=30 TH
EN  LET clo=0: BEEP .02,RND*-20
4097 IF m$="" THEN  GO TO 4090
4100 BEEP .002,44
4105 IF m$>"6" OR m$<"1" THEN  G
O TO 4080
4110 IF t(ca,3)<VAL m$ THEN  PRI
NT  INK opc+4;AT 21,0;"NO PUEDE
AVANZAR ";m$;" EN CASILLA ";ca:
BEEP .5,0: BEEP .09,-10: BEEP .3
,-2: GO TO 4000
4120 PRINT m$: LET av=VAL m$
4121 PRINT  INK opc+4;AT (opc=1)
+(opc=2)*14,(opc=1)+(opc=2)*22;"
JUGADOR ";opc
4130 FOR l=1 TO av
4140 LET cic=1: GO SUB 6000
4145 LET xan=x: LET yan=y
4150 IF opc=1 THEN  LET t(ca,1)=
t(ca,1)+1
4155 IF opc=2 THEN  LET t(ca,2)=
t(ca,2)-1
4160 LET cic=1: GO SUB 6000
4165 GO SUB 6500
4170 LET cic=2: GO SUB 6000
4180 NEXT l
4190 LET t(ca,3)=t(ca,3)-av
4200 IF t(ca,3)=0 THEN  LET ncab
lk=ncablk+1: GO SUB 8000
4210 IF ncablk=8 THEN  GO TO 700
0
4999 RETURN
5000 DIM t(8,3)
5005 LET ncablk=0
5020 FOR n=7 TO 0 STEP -1
5025 LET t(n+1,1)=1
```

```
5030 LET t(n+1,2)=8
5035 LET t(n+1,3)=6
5040 FOR m=0 TO 7
5050 IF n<>0 AND n<>7 THEN  PLOT
 3+n*16+m*16,39+n*16: DRAW 12,0:
 DRAW 8,8: DRAW -12,0: DRAW -8,-
8
5060 IF n=0 THEN  PRINT AT 15,1+
m*2;" ";AT 16,1+m*2;" "; INVER
SE 1;AT 18,m*2;m+1
5065 IF n=7 THEN  PRINT  INK 5;A
T 1,15+m*2;" ";AT 2,15+m*2;" "
;AT 0,15+m*2; INVERSE 1;m+1
5070 NEXT m
5090 NEXT n
5094 INK 5: PLOT 0,174: DRAW 86,
0: DRAW 0,-40: DRAW -86,0: DRAW
0,40
5095 PLOT 2,172: DRAW 82,0: DRAW
 0,-36: DRAW -82,0: DRAW 0,36
5096 INK 6: PLOT 168,70: DRAW 86
,0: DRAW 0,-40: DRAW -86,0: DRAW
 0,40
5097 PLOT 170,68: DRAW 82,0: DRA
W 0,-36: DRAW -82,0: DRAW 0,36
5099 RETURN
5100 RESTORE 5200
5105 FOR n=0 TO 31: READ a: POKE
 USR "a"+n,a: NEXT n
5199 RETURN
5200 DATA 0,20,25,26,58,206,31,1
15
5210 DATA 142,54,127,63,26,114,2
04,24
5220 DATA 0,36,21,26,28,72,41,62
5230 DATA 47,90,247,255,245,93,8
9,204
6000 LET y=(opc=1)*t(ca,1)*2+(op
c=2)*t(ca,2)*2-1
6010 LET x=ca*2-1
6020 LET x=x+(opc=1)*2*(8-t(ca,1
))+(opc=2)*2*(8-t(ca,2))
6040 FOR n=1 TO cic
6050 PRINT  OVER 1; INK opc+4;AT
 y,x;CHR$ (142+opc*2);AT y+1,x;C
HR$ (143+opc*2)
6055 BEEP .003,RND*60
6060 NEXT n
6099 RETURN
6500 FOR m=-1 TO 3
6510 PLOT xan*8+m,162-yan*8+m
6520 DRAW  OVER 1; INK opc+4;8,0
6590 NEXT m
6594 IF opc=1 AND t(ca,1)=2 THEN
  INK 5: PLOT xan*8+2,166-yan*8:
 DRAW -2,10: DRAW 8,0: DRAW 2,-1
0
6596 IF opc=2 AND t(ca,2)=7 THEN
  INK 6: PLOT xan*8-2,160-yan*8:
 DRAW -6,-9: DRAW 8,0: DRAW 6,9
6599 RETURN
7000 PRINT AT 21,0; FLASH 1;"GAN
A JUGADOR ";opc: PAUSE 140
7003 PAUSE 140
7005 LET tju1=tju1+(opc=1)
7006 LET tju2=tju2+(opc=2)
7010 CLS : PRINT AT 8,12;"MARCAD
OR";AT 12,2;"JUGADOR 1 - ";tju1;
"  JUGADOR 2 - ";tju2
7015 PRINT AT 15,3;"Pulsa BREAK
para continuar": PRINT USR 1314
7020 PAUSE 0
7030 GO TO 20
8000 LET y=169-t(ca,2)*16
8004 LET x=16*(ca+8-t(ca,2))-10
8020 PLOT  INK 2;x+2,y
8031 PLOT  INK 2;x+24,y+16
8099 RETURN
9000 LET ink=5: LET m$="       Do
s grupos de canguros se avistan
y corren a encontrarse.        "
9002 GO SUB 9100
9005 LET m$="       Como se sabe
 no demuestra buen comportamient
o  quien  mas prisa tenga."
9010 GO SUB 9100
9020 LET m$="      A pesar de la
 impaciencia,hay que procurar qu
e sea un can-guro de tu grupo  e
l  ultimo en reunirse con su cor
respondiente tocayo."
9030 GO SUB 9100
9040 LET m$="      Solo de esta
forma puedes ser considerado gan
ador.
        "



9050 GO SUB 9100
9060 LET m$="  PULSA .ENTER. PAR
A COMENZAR"
9070 GO SUB 9100
9080 PAUSE 0
9099 RETURN
9100 BEEP .004*RND,20*RND: PRINT
  INK ink;m$( TO 2);
9110 LET m$=m$(3 TO )
9120 IF LEN m$<3 THEN  PRINT m$:
 PRINT : LET ink=ink+1: RETURN
9130 GO TO 9100
9999 SAVE "Reunion K" LINE 1: GO
 TO 9999
```

Programación lineal

Dentro de los métodos de resolución de problemas de ingeniería, existe uno que es particularmente conocido por sus múltiples aplicaciones. Nos referimos a la programación lineal. Esta rama de la ciencia goza de gran aceptación entre los ingenieros y es el tema sobre el que versa el programa que nos envía Juan Luis Martínez.
El programa combina tres métodos para la resolución de problemas de maximización de resultados. La matriz x(n) es el vector de variables de decisión, c(n) es el vector de coeficientes de recursos, b(m) el vector del segundo miembro de las restricciones y a(m, n) es la matriz de los coeficientes de las restricciones o matriz tecnológica.
Los métodos que se combinan son el símplex, el dual y un método de programación entera.

```
  10 PAPER 6: INK 0: BORDER 6:
CLS : POKE 23609,100
  20 PRINT AT 6,6; PAPER 2; INK
7; FLASH 1;"PROGRAMACION LINEAL"
: PLOT 47,119: DRAW 153,0: DRAW
0,9: DRAW -153,0: DRAW 0,-9
  30 PRINT AT 10,7; PAPER 2; INK
 7; FLASH 1;"Pare  el cassette":
 PLOT 55,87: DRAW 137,0: DRAW 0,
9: DRAW -137,0: DRAW 0,-9
  40 PRINT AT 14,8; PAPER 2; INK
 7; FLASH 1;"Pulse una tecla": P
LOT 63,55: DRAW 121,0: DRAW 0,9:
 DRAW -121,0: DRAW 0,-9
  50 PRINT AT 20,5; Martinez G
omez 1985": PAUSE 0: CLS
  60 GO SUB 180: PRINT AT 3,0;"
Este programa combina el  algo-
ritmo del simplex, el dual y un
metodo  de  programcion  entera
para resolver problemas  cuya -
expresion general sea:"
  70 PRINT AT 9,0;"  Max c(1)x1+
c(2)x2+...+c(n)xn"
  80 PRINT AT 11,0;" s.a a(1,1)x
1+...+a(1,n)xn<=b(1)      a(2,1)x
1+...+a(2,n)xn<=b(2)            .
                 .              .
                 .              .
                 .         a(m,1)x
1+...+a(m,n)xn<=b(m)"
  90 PRINT AT 18,0;"    x(i)>=0
  i=1,2,...,n": GO SUB 190
 100 GO SUB 180: PRINT AT 3,0;"N
ota: Los coeficientes se han de
introducir con su correspondien
te signo.                      E
jemplo:

       Max   -x(1)+2x(2)

          s.a  -x(1)+x(2)<=2
                3x(1)-x(2)<=1


              x(1)>=0, x(2)>=0"
 110 PRINT AT 15,0;"  c(1)=-1  b
(1)= 2  a(1,1)=-1     c(2)= 2  b
(2)= 1  a(1,2)= 1
        a(2,1)= 3
        a(2,2)=-1": GO SUB 190
 120 GO SUB 180: PRINT AT 5,0;"
Al aplicar este  metodo  apare-
cen n+m soluciones,desde la n+1
hasta la n+m son las  denomina-
das variables de holgura, que -
corresponde una por  cada  res-
triccion.

Al completar cada restriccion -
con una nueva variable,estas ya
no se cumplen como  restriccio-
nes de desigualdad, sino como -
restricciones de igualdad.": GO
SUB 190: GO TO 200
 130 INPUT "Deseas instrucciones
 (s/n) ? ";a$
 140 IF a$="s" THEN  GO TO 60
 150 IF a$<>"s" THEN  GO TO 200
 180 PRINT AT 1,6; PAPER 2; INK
7; FLASH 1;"PROGRAMACION LINEAL"
: PLOT 47,159: DRAW 153,0: DRAW
0,9: DRAW -153,0: DRAW 0,-9: RET
URN
 190 PRINT AT 20,8; PAPER 2; INK
 7; FLASH 1;"Pulse una tecla": P
LOT 63,7: DRAW 121,0: DRAW 0,9:
DRAW -121,0: DRAW 0,-9: PAUSE 0:
 CLS : RETURN
 199 REM *** entra datos ***
 200 INPUT "Numero de incognitas
 n=";n: PRINT "Numero de incogni
tas n=";n''"Correcto ? (s/n)": P
AUSE 0: CLS : IF INKEY$="n" THEN
  GO TO 200
 210 INPUT "Numero de restriccio
nes m=";m: PRINT "Numero de rest
ricciones m=";m;''"Correcto ? (s
/n)": PAUSE 0: CLS : IF INKEY$="
n" THEN  GO TO 210
```

```
PROGRAMACION LINEAL

Este programa combina el algo-
ritmo del simplex, el dual y un
metodo de programcion entera
para resolver problemas cuya -
expresión general sea:

 Max c(1)x1+c(2)x2+...+c(n)xn

s,a a(1,1)x1+...+a(1,n)xn<=b(1)
    a(2,1)x1+...+a(2,n)xn<=b(2)
        .                    .
        .                    .
        .                    .
    a(m,1)x1+...+a(m,n)xn<=b(m)

  X(i)>=0      i=1,2,...,n

     Pulse una tecla
```

```
 220 IF m<2 THEN  PRINT AT 10,0;
" El numero de ecuaciones  m  ha
  de  se  mayor  o  igual  que 2
  (m>=2).": PAUSE 300: CLS : GO
TO 210
 230 DIM a(m,n): DIM c(n): DIM b
(m)
 240 PRINT "  Introduce coeficie
ntes de"''"  recursos; c(n)."
 250 FOR j=1 TO n
 260 INPUT ("c(";j;")=");c(j)
 270 NEXT j
 280 CLS : PRINT "Introduce  coe
ficientes  segundo"''"miembro re
stricciones; b(m)."
 290 FOR i=1 TO m
 300 INPUT ("b(";i;")=");b(i)
 310 NEXT i
 320 CLS : PRINT "  Introduce co
eficientes"''"  tecnicos; a(m,n)
."
 330 FOR i=1 TO m: FOR j=1 TO n
 340 INPUT ("a(";i;",";j;")=");a
(i,j)
 350 NEXT j: NEXT i
 360 CLS : PRINT  PAPER 7;" Coef
icientes de recursos c(n): ": PR
INT
 370 FOR j=1 TO n: PRINT "c(";j;
")=";c(j): NEXT j
 380 PRINT #0; PAPER 7;" Para co
rregir errores pulse c   Cualqui
er  otra  para  seguir  ": PAUSE
 0
 390 IF INKEY$="c" THEN  INPUT "
Numero coeficiente erroneo ";j:
INPUT ("Introduce nuevo coeficie
nte      c(";j;")=");c(j): GO TO
360
 400 CLS : PRINT  PAPER 7;" Coef
icientes de recursos b(m): ": PR
INT
 410 FOR i=1 TO m: PRINT "b(";i;
")=";b(i): NEXT i
 420 PRINT #0; PAPER 7;" Para co
rregir errores pulse c   Cualqui
er  otra  para  seguir  ": PAUSE
 0
```

```
 430 IF INKEY$="c" THEN  INPUT "
Numero coeficiente erroneo ";i:
INPUT ("Introduce nuevo coeficie
nte     b(";i;")=");b(i): GO TO
400
 440 CLS : PRINT  PAPER 7;" Coef
icientes tecnicos a(m,n):  ": PR
INT
 450 FOR i=1 TO m: FOR j=1 TO n:
 PRINT "a(";i;",";j;")=";a(i,j):
 NEXT j: NEXT i
 460 PRINT #0; PAPER 7;" Para co
rregir errores pulse c   Cualqui
er  otra  para  seguir  ": PAUSE
 0
 470 IF INKEY$="c" THEN  INPUT "
Subindices coeficiente erroneo "
,i,j: INPUT ("Introduce nuevo co
eficiente     a(";i;",";j;")=");
a(i,j): GO TO 440
 480 GO SUB 980
 499 REM *** tabla inicial ***
 500 DIM s(m+1,m+n+1)
 510 FOR i=1 TO m+1: FOR j=1 TO
n+m+1
 520 IF i<m+1 AND j<n+1 THEN  LE
T s(i,j)=a(i,j)
 530 IF i<m+1 AND (n<j<m+n+1) AN
D j-i=n THEN  LET s(i,j)=1
 540 IF j=m+n+1 AND i<m+1 THEN
LET s(i,j)=b(i)
 550 IF i=m+1 AND j<n+1 THEN  LE
T s(i,j)=-c(j)
 560 NEXT j: NEXT i
 590 LET cs=0
 599 REM *** algoritmo ***
 600 FOR j=1 TO m+n
 610 IF s(m+1,j)<0 THEN  GO TO 1
000
 620 NEXT j
 630 FOR i=1 TO m
 640 IF s(i,m+n+1)<0 THEN  GO TO
 2000
 650 NEXT i
 655 GO TO 900
 660 DIM x(n+m)
 670 FOR j=1 TO m+n: LET cont=0:
 LET cont1=0
 680 FOR i=1 TO m+1
 690 IF s(i,j)=1 THEN  LET cont=
cont+1: LET f=i
 700 IF s(i,j)=0 THEN  LET cont1
=cont1+1
 710 NEXT i
 720 IF cont=1 AND cont1=m THEN
 LET x(j)=s(f,m+n+1)
 730 NEXT j
 740 CLS : PRINT AT 0,0; PAPER 7
;" Las soluciones al problema qu
e  me ha planteado son :
  ": PRINT
 750 FOR i=1 TO n+m
 760 PRINT TAB 3;"x(";i;")=";x(i
)
 770 NEXT i
 780 PRINT : PRINT  PAPER 7;" El
 valor de la funcion objetivo es
 :                            ":
PRINT : PRINT TAB 8;s(m+1,m+n+1)
 785 GO SUB 950
 790 CLS : PRINT #0; PAPER 7;" P
ara ver ultima tabla pulse t   P
ara empezar pulse e            P
ara terminar pulse f           P
ara p. entera pulse p          ":
 PAUSE 0
 800 IF INKEY$="e" THEN  CLS : G
O TO 130
 810 IF INKEY$="f" THEN  STOP
 820 IF INKEY$="t" THEN  GO TO 8
40
 825 IF INKEY$="p" THEN  GO TO 3
000
 830 IF INKEY$<>"e" OR INKEY$<>"
f" OR INKEY$<>"t" THEN  GO TO 80
0
 840 CLS : PRINT AT 0,0; PAPER 7
;"          Ultima tabla:
  ": PRINT
 850 FOR i=1 TO m+1: FOR j=1 TO
m+n+1
 860 PRINT TAB 3;"s(";i;",";j;")
=";s(i,j)
 870 NEXT j: NEXT i
 880 GO SUB 950
 890 GO TO 790
 900 LET cont1=0: FOR j=1 TO m+n
 910 IF s(m+1,j)=0 THEN  LET con
t1=cont1+1
 920 NEXT j
 930 IF cont1>m THEN  CLS : PRIN
T AT 10,0;" El problema posee mu
ltiples so- luciones. He calcula
do  una  de ellas.": GO SUB 950
 940 GO TO 660
 950 PRINT #0; PAPER 7;"
 Pulse una tecla         ": PAUSE
 0: RETURN
 960 IF cs=2 THEN  CLS : PRINT A
T 9,0;"No puedo continuar ni apl
icando el algoritmo de simplex
ni  el algoritmo dual, la soluci
```

Todospectrum

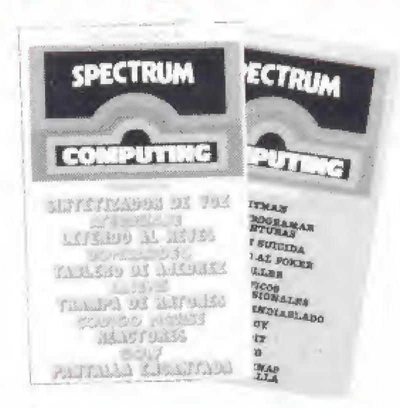


TODOSPECTRUM es una publicación mensual que le ayudará a obtener el máximo partido a su **SPECTRUM** y al **ZX 81**.



CONOZCA LAS VENTAJAS DE SUSCRIBIRSE A

Todospectrum



ADEMAS, le hacemos un 25 % DE DESCUENTO

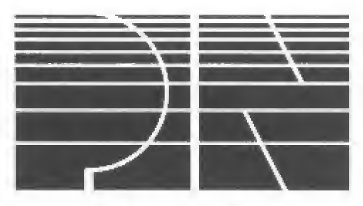
sobre el precio real de suscripción (12 números)

VALOR REAL DE SUSCRIPCION	OFERTA ESPECIAL DE SUSCRIPCION	USTED AHORRA
~~3.600~~ PTAS.	2.700 PTAS.	900 PTAS.

APROVECHE AHORA esta oportunidad irrepetible para suscribirse a **TODOSPECTRUM**. Envie **HOY MISMO** la tarjeta adjunta a la revista, que no necesita sobre ni franqueo. Deposítela en el buzón más cercano. Inmediatamente recibirá su primer ejemplar de **TODOSPECTRUM** más el **REGALO**.

Todospectrum
Bravo Murillo, 377
Tel. 733 79 69
28020 MADRID

```
on  es ilimitada.": GO SUB 950:
GO TO 790
 970 RETURN
 980 CLS : PRINT AT 9,5; PAPER 2
; INK 7; FLASH 1;"Realizando ope
raciones": PLOT 39,95: DRAW 177,
0: DRAW 0,9: DRAW -177,0: DRAW 0
,-9
 990 PRINT AT 11,7; PAPER 2; INK
 7; FLASH 1;"Espere 'un momento"
: PLOT 55,79: DRAW 145,0: DRAW 0
,9: DRAW -145,0: DRAW 0,-9: PRIN
T AT 20,5; Martinez Gomez 1985
"
 995 RETURN
 999 REM *** simplex ***
1000 LET min=s(m+1,1): LET c=1
1010 FOR j=2 TO m+n
1020 IF min>s(m+1,j) THEN  LET m
in=s(m+1,j): LET c=j
1030 NEXT j
1040 LET s=0: FOR i=1 TO m
1050 IF s(i,c)>0 THEN  LET min1=
s(i,m+n+1)/s(i,c): LET s=s+1: LE
T f=i: LET i=m
1060 NEXT i
1070 IF s=0 THEN  LET cs=cs+1: G
O SUB 960: GO TO 2000
1080 FOR i=2 TO m
1090 IF s(i,c)<=0 THEN  GO TO 11
10
1100 IF s(i,m+n+1)/s(i,c)<min1 T
HEN  LET min1=s(i,m+n+1)/s(i,c):
 LET f=i
1110 NEXT i
1120 LET ope=s(f,c)
1130 FOR j=1 TO m+n+1
1140 LET s(f,j)=s(f,j)/ope
1150 NEXT j
1160 FOR i=1 TO m+1
1170 IF i<>f THEN  LET ope=-(s(i
,c)*s(f,c))
1180 IF i=f THEN  GO TO 1220
1190 FOR j=1 TO m+n+1
1200 LET s(i,j)=s(f,j)*ope+s(i,j
)
1210 NEXT j
1220 NEXT i: LET cs=0
1230 GO TO 600
1999 REM *** dual ***
2000 LET min=s(1,m+n+1): LET f=1
2010 FOR j=2 TO m
2020 IF min>s(j,m+n+1) THEN  LET
 min=s(j,m+n+1): LET f=j
2030 NEXT j
2040 LET s=0: FOR i=1 TO m+n
2050 IF s(f,i)<0 AND s(m+1,i)>=0
 THEN  LET min1=ABS (s(m+1,i)/s(
f,i)): LET s=s+1: LET c=i: LET i
=m+n
2060 NEXT i
2070 IF s=0 THEN  LET cs=cs+1: G
O SUB 960: GO TO 1000
2080 FOR i=2 TO n
2090 IF s(f,i)>=0 THEN  GO TO 21
10
2100 IF ABS (s(m+1,i)/s(f,i))<mi
n1 THEN  LET min1=ABS (s(m+1,i)/
s(f,i)): LET c=i
2110 NEXT i
2120 GO TO 1120
2999 REM *** p. entera ***
3030 CLS : GO SUB 980
3040 DIM r(m)
3050 FOR i=1 TO m
3060 LET r(i)=s(i,m+n+1)-INT s(i
,m+n+1)
3070 NEXT i
3080 LET max=r(1): LET f=1
3090 FOR i=2 TO m
3100 IF max<r(i) THEN  LET max=r
(i): LET f=i
3110 NEXT i
3120 LET m=m+1
3130 DIM a(m+1,m+n+1)
3140 FOR j=1 TO m+n-1
3150 IF s(f,j)<>INT s(f,j) AND s
(f,j)<0 THEN  LET a(m,j)=-(1+s(f
,j))
3160 IF s(f,j)<>INT s(f,j) AND s
(f,j)>0 THEN  LET a(m,j)=-s(f,j)
3170 NEXT j
3180 LET a(m,m+n)=1: LET a(m,m+n
+1)=-max
3190 FOR i=1 TO m-1
3200 FOR j=1 TO m+n-1
3210 LET a(i,j)=s(i,j)
3220 NEXT j
3230 LET a(i,m+n+1)=s(i,m+n)
3240 NEXT i
3250 FOR j=1 TO m+n-1
3260 LET a(m+1,j)=s(m,j)
3270 NEXT j
3280 LET a(m+1,m+n+1)=s(m,m+n)
3290 DIM s(m+1,m+n+1)
3300 FOR i=1 TO m+1
3310 FOR j=1 TO m+n+1
3320 LET s(i,j)=a(i,j)
3330 NEXT j: NEXT i
3340 GO TO 600
9000 SAVE "P. Lineal" LINE 10: V
ERIFY "P. Lineal"
```



PIN SOFT, S.A.

Paseo de Gracia, 11 - Esc. C., 2º 4.ª
Tel. (93) 318 24 53 - 08007 Barcelona

SOFTWARE SPECTRUM

S.I.T.I. V.3* — **4.000**
Base de datos con cálculos.
Al comprar esta versión abonamos 3.000,- ptas. por cualquier versión anterior.

Context V.9* — **4.000**
Tratamiento de Textos.
Funciona con cualquier impresora.
Acentos graves y agudos.
Al comprar esta versión abonamos 3.000 ptas. por cualquier versión anterior.

Adaptador SITI-CONTEXT — **2.500**
Permite pasar información del SITI al CONTEXT.

M.D.S. - Sistema Operativo para Microdrive — **7.000**
Conjunto de nuevos comandos BASIC que permiten Acceso Aleatorio a Ficheros en Microdrive con un tiempo medio de acceso de 4 segundos.

CONTABILIDAD PIN* — **3.000**
Plan contable 200 cuentas. 2.000 asientos.
Hasta 9.000.000.000 Balance con activo-pasivo, cta. resultados. Utiliza el S.O.M.D.S. Cualquier impresora 80 col.

Kit Utilidades Discovery 1 — **3.000**
10 utilidades CAT extendido. ON ERROR. Compactador de discos, etc.

AJUSTE DE CABEZALES CASSETTE	2.500
SINTETIZADOR DE VOZ	3.000
MULTI-COPYS (Copys desde 2 cm. hasta 70 cm.)	3.000
COPY GRISES (F+, SP-800, SP-1000, GP-550)	2.500
COPY RS-232	2.500
COPY SERIE RITEMAN F+	2.500
EDITOR 64 (64 columnas en pantalla)	2.750

***Disponible en disco para Discovery 1 al precio de 5.000,- ptas.**

VIDEOJUEGOS

COMANDO	2.181
CRITICAL MASS	1.696
DAM BUSTERS	1.964
FIGHTING WARRIOR	1.875
GYROSCOPE	1.696
HIGHWAY ENCOUNTER	1.696
OLE TORO	1.875
RAMBO	1.875
SABOTEUR	1.696
SGRIZAM	1.741
SUPERMAN	2.464
THE WAY OF EXPLODING FIST	2.054
WEST BANK	1.741

Y TODAS LAS NOVEDADES.
Solicita catálogo.

NOVEDADES

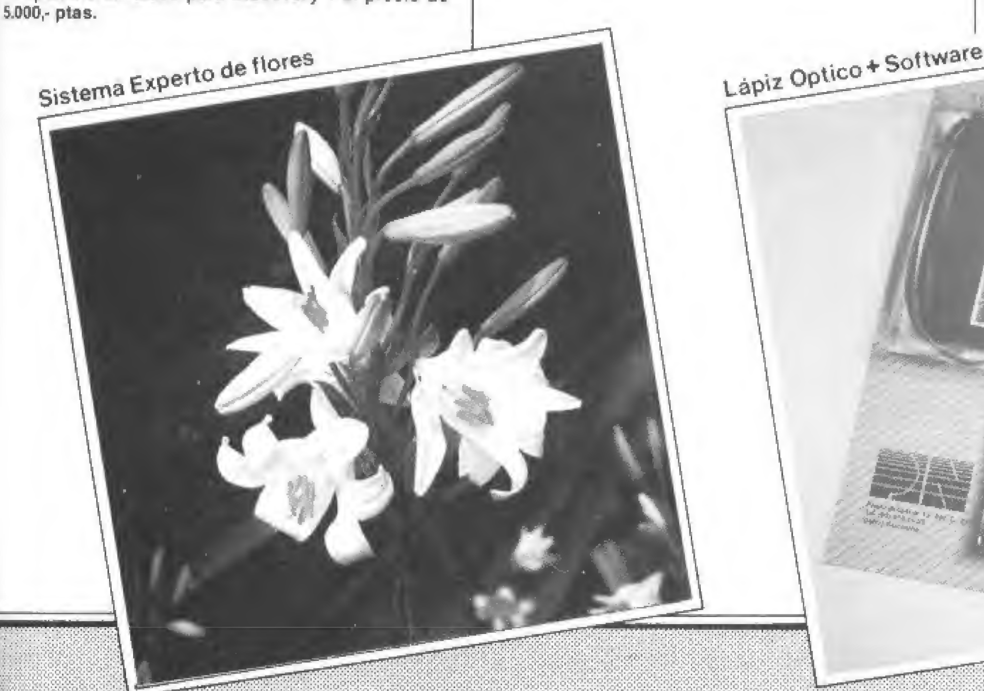
Sistema experto de Flores de jardín	3.500
Sistema experto de Minerales	3.500
APLICACIONES SITI V.3	3.500

Agenda+Vídeos+Contabilidad doméstica+Stocks, etc. (necesita el SITI V.3)

HARDWARE SPECTRUM

Interface sonido TV	3.500
Interface Joystik	2.000
Joystick Quickshot II	2.600
I/F Centronics	8.000
Lápiz óptico+software	4.850
Interface monitor	3.900
Cinta virgen 15'	100
Monitor Ciaegi f. verde	24.000
Monitor Ciaegi f. ámbar	24.750
Caja para 12 microdrives	100
Teclado Saga 1	11.000
Teclado Saga 3	19.900
Discovery 1+disco Kit	55.000
Diskettes 3 1/2	714
Cable impresora Discovery	3.500
Alimentación ininterrumpida	9.750
Digitalizador de imágenes	35.000
Impresora Riteman F+	71.900

Sistema Experto de flores



Lápiz Optico + Software

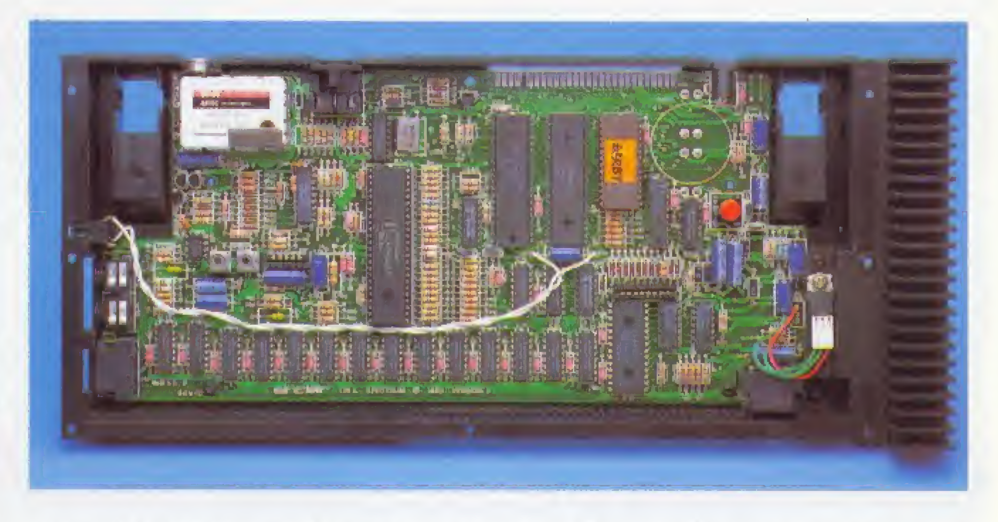
TIENDA AL PUBLICO
EN EL CENTRO DE BARCELONA
HORARIO: de 10 h a 20 h. ININTERRUMPIDO
SABADOS CERRADO

PEDIDOS POR CORREO O TELEFONO
Envíos contra reembolso a toda España
200 ptas. gastos de envío
En tu domicilio en 3-4 días

ANALISIS

GENERADOR

Cuando los fanáticos de conocer las cosas por dentro abrimos el Spectrum 128, aparte de no sacar nada en claro, vemos algunos circuitos integrados que se encuentran en el Spectrum de 48 *Kbytes.* Uno de ellos destaca por su tamaño: el AY-3-8912, el músico del 128.



El AY-3-8912 se denomina PSG, siglas inglesas de *Programable Sound Generator* (Generador Programable de Sonido) que fue el nombre que dio su primer fabricante **General Instruments,** al AY-3-8912 y a su hermano mayor el AY-3-8910.

El destino inicial de este circuito era la generación de efectos especiales en máquinas de marcianitos y la producción de sonido en órganos electrónicos. Posteriormente, ha demostrado su utilidad para sonorizar diversos ordenadores y actualmente, es raro el ordenador personal para el que no se haya editado alguna vez un artículo llamado algo así como «Haga música con su XXX». Montaje utilizando el GI AY-3 8912».

La importancia que tiene y ha tenido este dispositivo en el mundo informático se debe a la flexibilidad de su diseño, que impide que se quede obsoleto.

A continuación vamos a describir la función de cada una de las patas del PSG.

DA0-DA7: forman el *bus* de datos para comunicarse con el microprocesador. Sirven para seleccionar el registro de control a modificar y para enviar o recibir su valor, según el protocolo que veremos más adelante.

A8: sirve exclusivamente para añadir un *bit* de identificación del PSG en sistemas que dispongan de más periféricos. Para seleccionar el PSG deberemos poner A8 a nivel lógico «1» (+5 voltios).

BDIR, BC2, BC1: estas líneas indican al PSG el significado del valor puesto en DA0-DA7 o que se quiere leer de él, según la tabla 1. De este modo, a cambio de mayor complejidad interna del *chip,* conseguimos una gran simplificación tanto de la circuitería exterior como del soporte lógico *(software)* de funcionamiento del equipo.

IOA0-IOA7: son las líneas externas del puerto de entrada y salida A, que analizaremos más adelante.

TEST1: Es un punto que utiliza el fabricante para comprobar el buen funcionamiento del PSG antes de ponerlo a la venta. La única recomendación que **General Instruments** hace para su buen funcionamiento es que se deje sin conectar a absolutamente nada.

CLOCK: Entrada de reloj tipo TTL, utilizada como base de tiempos para la generación de tonos, envolventes y ruidos.

Vcc, Vss: Tensiones de alimentación +5 y 0 voltios, respectivamente.

Lectura y modificación de registros

Vamos a estudiar ahora cómo dar valores a los registros y leer.

Ante todo analizaremos el sistema general de escritura en el PSG. El sistema será: a) poner el valor a escribir en

I S D E L
DE SONIDOS

DA0-DA7; b) poner los valores adecuados en BDIR, BC1 y BC2 (típicamente BDIR=1, BC2=2, y BC1=1 para escribir una dirección y BC=0 para dar valor a un registro).

Para leer un valor, habrá que poner primero los valores adecuados en BDIR (0), BC1 (1) y BC1 (1) y a continuación leer el dato en DA0-DA7.

De este modo, el proceso completo para dar un valor a un registro es a) realizar una operación de escritura con BC1=1 y DA0-DA3 con el número binario que represente el número de registro a modificar. A continuación, realizaremos otra operación de escritura con BC1=0, y DA0-DA7 con el valor a meter en el registro.

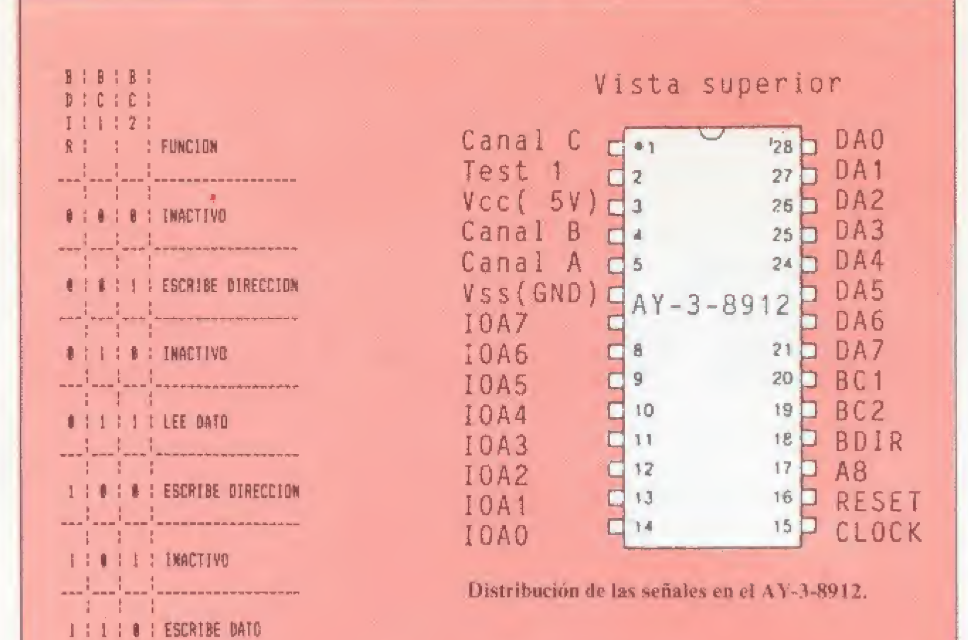
Para leer el valor de un registro, se deberá hacer una operación de escritura como la primera que se realizó para escribir en un registro, y a continuación una operación de lectura, leyendo en DA-DA7 el valor del registro pedido.

Generación de tonos

La generación de tonos se realiza mediante dos contadores. El primero de ellos toma como valor inicial el número de 12 *bits* contenido en el registro de control de tono del canal correspondiente (R0 y R1 para el A, R2 y R3 para el B, R4 y R5

El chip AY-3-8910 es el responsable de las increíbles posibilidades musicales del MSX

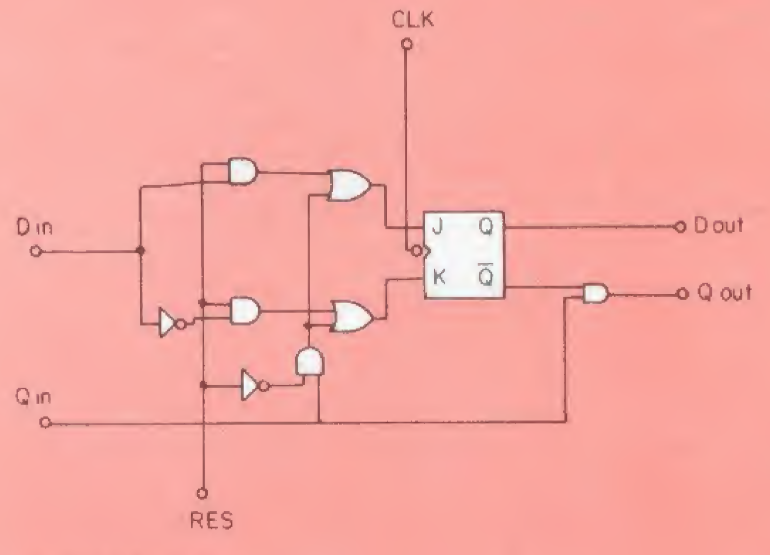




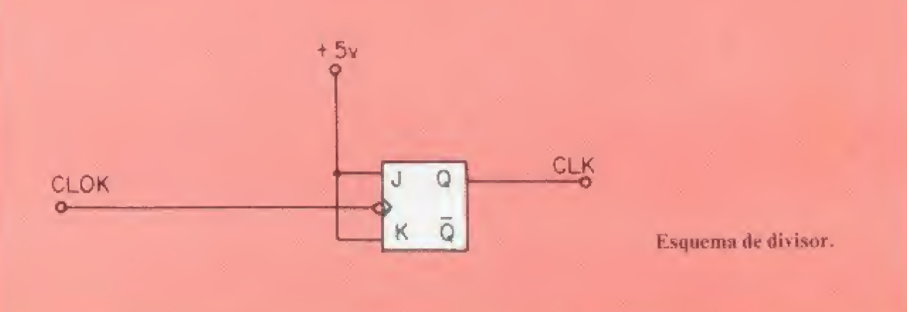
BDIR	BC1	BC2	FUNCION
0	0	0	INACTIVO
0	0	1	ESCRIBE DIRECCION
0	1	0	INACTIVO
0	1	1	LEE DATO
1	0	0	ESCRIBE DIRECCION
1	0	1	INACTIVO
1	1	0	ESCRIBE DATO
1	1	1	ESCRIBE DIRECCION

Tabla 1: Modos de transferencia seleccionados según el estado de BDIR, BC1 y BC2.

Distribución de las señales en el AY-3-8912.



Esquema del bloque base.



Esquema de divisor.

para el C) y va quitándole 1 para cada dos ciclos de reloj externo *(CLOCK)*. Cada vez que este contador pasa por 0, vuelve a tomar el valor de registro de control y genera un pulso.

Dicho pulso se utiliza como entrada para el siguiente contador, que es de dos *bits* y por tanto puede tomar 4 valores distintos (00, 01, 10 y 11). Este contador suma 1 cada vez que recibe un impulso. Cuando llega a 11, repite este valor y luego comienza a restar 1 hasta llegar a 00, valor que también repite para luego comenzar a sumar 1, repitiéndose el ciclo hasta que se ponga 0 en el registro de control de tono.

Si tomamos los dos bits del último contador y los utilizamos como entrada de un conversor analógico digital, conseguiremos una señal que varía periódicamente en el tiempo con la frecuencia dada por la siguiente fórmula:

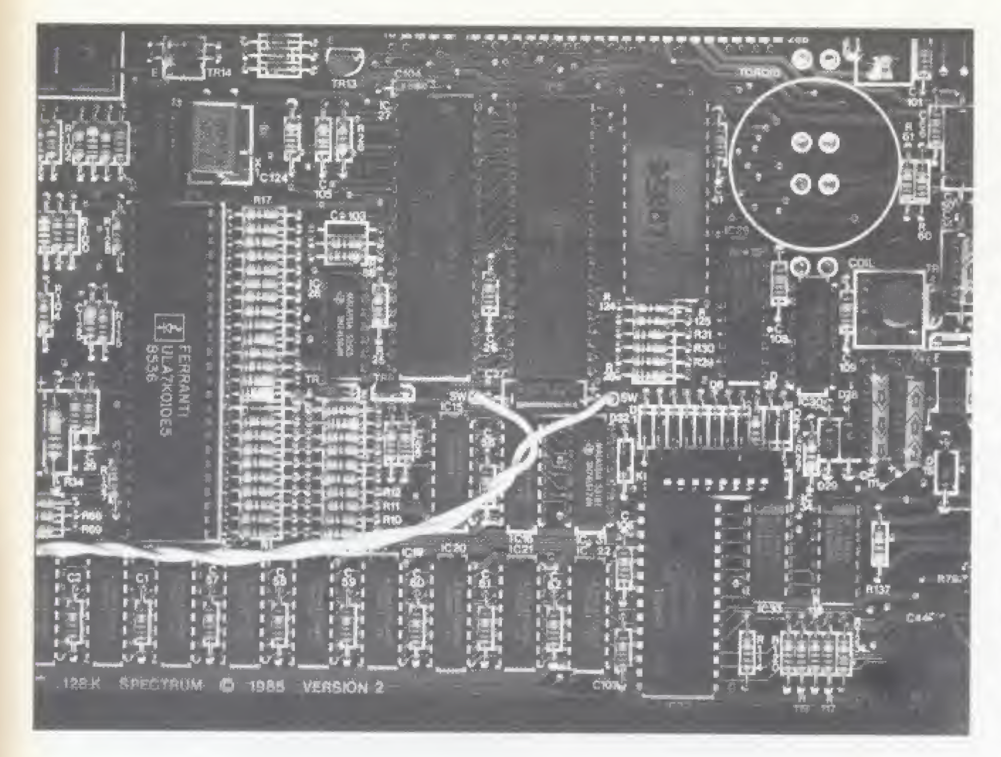
$$f = \frac{fr}{16*tp}$$

f = frecuencia de la señal de salida
fr = frecuencia de reloj
tp = valor contenido en el registro de control de tono. En el canal A, sería R0 + 256* (valor contenido en los bits 0 a 3 en R1).

Así pues, para conseguir una frecuencia de aproximadamente 440 Hz (la central del piano), habrá que poner en el registro de control un valor de 1789772.5/(16*440) = 254, es decir, poner en R0 el valor 254 y en R1 0.

Generación de ruido

La generación de ruidos se realiza por un método similar, sólo que ahora el valor inicial que toma el contador es el que hay en el registro R6, en los *bits* 0 ó 4, poniéndose ceros en los *bits* superiores. Con la señal que así se genera, se modela la amplitud de un generador de



ruido blanco. Hay que recordar un detalle importante: el canal de ruido es único, y lo que hacemos es decir (mediante R7, como veremos después) en qué canales queremos oírlo.

Generación de envolventes

La generación de envolventes de volumen se controla mediante dos registros diferentes. La forma la controla el registro R13 y la duración de la rampa los registros R11 y R12.

El registro R13 de forma de envolvente sólo utiliza los 4 bits de menos peso, llamados *CONT, ATT, ALT* y *HOLD,* cuyas funciones son:

HOLD (Mantener): si está puesto a nivel lógico «1», al acabar el primer ciclo de variación de la envolvente, mantendrá el valor con el que se acabe la rampa (volumen máximo en caso de rampa ascendente o silencio en caso de rampa descendente). En caso de valer «0» la envolvente se repetirá cíclicamente.

ALT (Alternancia): en caso de que valga «1», se alternan rampas ascendentes y descendentes.

ATT (Ataque): si vale «1», la envolvente comenzará con una rampa ascendente, en caso contrario empezará con una rampa descendente.

CONT (Continuación): si vale «1», la forma de la envolvente será la definida por el *bit HOLD.* Si no, después del primer ciclo se volverá al estado de silencio.

La duración de la primera rampa (y de las siguientes, si las hubiere) viene definida por los registros R11 y R12, según la fórmula:

$$T = 256\frac{256 * R12 + R11}{fr}$$

Donde fr es la frecuencia de reloj.

Así, para conseguir que un sonido crezca desde el silencio hasta el volumen máximo durante 1 segundo, y luego calle por completo, habrá que poner el valor 4 en R13, el valor 27 en R12 y 79 en R11.

El sistema utilizado para generar dichas envolventes es muy similar al de generación de tonos, con algunas diferencias:

a) El primer contador es de 16 bits en vez de 12.

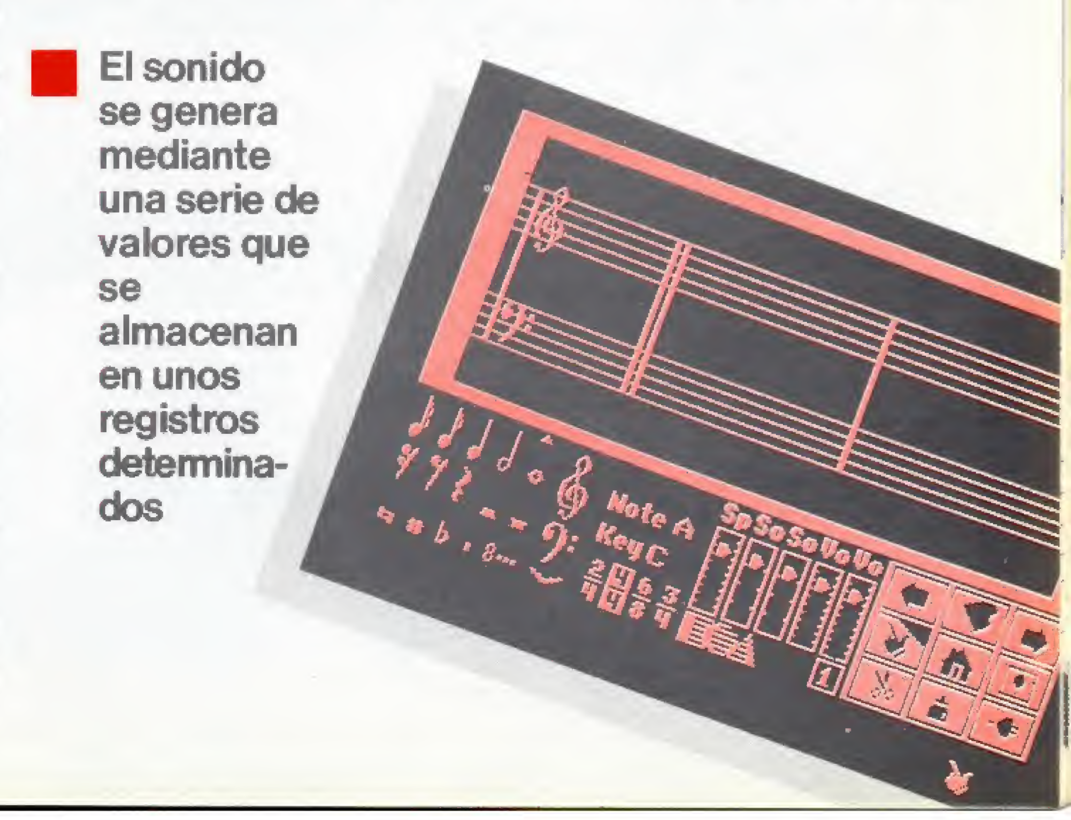
b) El reloj que se introduce en el primer contacto no se divide previamente por 2.

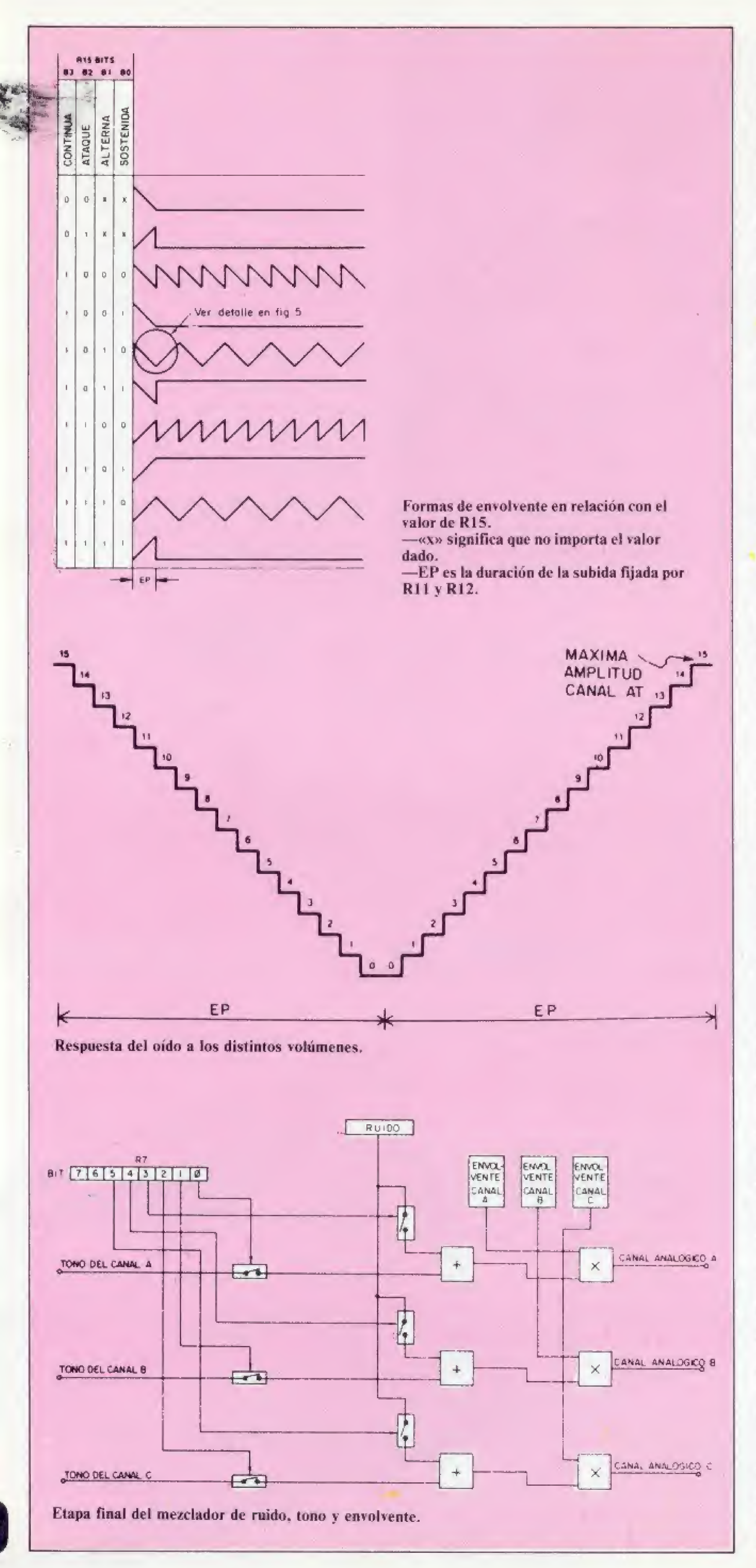
c) El segundo contador es de 4 bits, lo que genera 16 niveles distintos, y es mucho más complicado para poder ser controlado por los *bits HOLD, ALT, ATT* y *CONT.*

d) El convertidor de digital a analógico no es lineal, como el de los tonos, sino exponencial, para contrarrestar la respuesta logarítmica de nuestro oído.

Para que un canal quede afectado por el generador de en-

El sonido se genera mediante una serie de valores que se almacenan en unos registros determinados





Formas de envolvente en relación con el valor de R15.
—«x» significa que no importa el valor dado.
—EP es la duración de la subida fijada por R11 y R12.

Respuesta del oído a los distintos volúmenes.

Etapa final del mezclador de ruido, tono y envolvente.

volvente, habremos de poner el *bit* 4 de su registro (R8 para el canal A, R9 para el canal B y R10 para el canal C) de control envolvente a «a». Caso contrario, su volumen será el fijado por los *bits* 0 ó 3 de dicho registro, que sustituirán a la entrada del convertidor D/A de envolvente a los 4 *bits* del segundo contador.

Mezcla final de sonidos

Hasta ahora hemos visto cómo se genera cada parte del sonido. Ahora nuestro problema es entender cómo se unen dichas partes para crear el sonido definitivo.

Analicemos, pues, la naturaleza de lo que queremos oír. Nuestra idea principal es percibir un tono base. A este tono podemos añadirle un ruido, lo que significa SUMARLE un ruido. Cuando hablamos de volumen, nos referimos a un valor máximo que pueden alcanzar los picos de nuestra señal de tono, es decir, se trata de una MULTIPLICACION del tono más el ruido por la envolvente de volumen (o el volumen fijo).

Los puertos de entrada y salida

Centrándonos en el Z-80, que es el microprocesador que lleva el Spectrum, hemos de resaltar que tiene dos espacios de direcciones diferenciados. Uno, que funciona más rápido, se denomina clásicamente el espacio de MEMORIA. A él se accede por medio de los 16 *bits* del *bus* de direcciones y por la señal de control *MREQ*. Es, pues, un espacio de 65536 lugares posibles. El otro espacio, al que se accede por medio de sólo 8 *bits* del *bus* de direcciones y de la señal de control IOREQ, es el llamado espacio de E/S.

Es en este segundo espacio donde se sitúan los periféricos. Pero existen varios problemas graves: ni todos los periféricos entienden lo que son las señales

de control, ni todos se pueden conectar a un *bus* de datos, ni todos van tan rápido como el procesador, ni todos entienden las informaciones en binario. Para poder dialogar con este tipo de periféricos se inventaron los puertos e/s, que básicamente son lugares donde se deja un dato hasta que pueda recogerlo el periférico o hasta que se convierta a un formato por él comprensible, o donde se recoge un dato cuando el periférico ha podido elaborarlo. En sí mismos son memorias de pequeño tamaño (1 octeto, 1/2 octeto) que están conectadas a dos *buses*, el bus de datos del microprocesador y el bus del periférico, llamado normalmente «*bus* externo».

Realizada la disquisición filosófica, pasemos a ver el funcionamiento del puerto del AY-3-8912, denominado A. Los datos intercambiados con el periférico se leen y escriben en el registro R14 (puerto A).



Una serie de registros permiten controlar todas las funciones necesarias de sonido

Este registro sólo puede funcionar de dos modos: o sólo entrada o sólo salida. El hecho de que funcione en uno u otro modo viene determinado por unos bits del registro de control (R7).

El registro general de control R7

El registro R7 cumple la función de controlar la mezcla de ruidos y tonos que se realiza, además de fijar la dirección de funcionamiento del puerto e/s. Para ello está dividido en tres bloques según se ve en la figura 8: TONO, RUIDO y ENTRADA/SALIDA; los dos primeros formados por tres *bits* (uno por canal) y el último por dos *bits*, uno por puerto.

Los *bits* 0 a 2 controlan el tono de cada canal del modo siguiente: si valen «1», en su canal correspondiente no sonará el tono fijado por los registros de tono, en caso contrario sí lo hará.

Los bits 3 a 4 controlan el ruido. Caso de valer «0», se sumará el ruido al tono del canal.

Los *bits* 6 a 7 fijan la dirección de funcionamiento del puerto. Si uno de estos bits vale «1», cada vez que se envíe un dato del procesador al puerto, éste «memorizará» el valor y lo mantendrá en las líneas del bus exterior de datos.

Juan Zarategui

PROTEJA SU SPECTRUM PLUS CON ESTA PRACTICA FUNDA

A UN PRECIO ESPECIAL

OFERTA LIMITADA
Y EXCLUSIVA PARA
NUESTROS LECTORES

AHORA PARA USTED 975 PTAS.

ZX

Aproveche la oportunidad de mantener como nuevo su Spectrum Plus con esta funda, y beneficiese de un 30% de descuento sobre su precio normal.

¡APRESURESE! RECORTE Y ENVIE HOY MISMO ESTE CUPON A:
PUBLINFORMATICA (Dto. FUNDAS), C/BRAVO MURILLO, 377 5.° A 28020 MADRID

CUPON DE PEDIDO

Sí, envíeme al precio de 975 Ptas. cada una, ____ fundas para mi SPECTRUM PLUS.
El importe lo abonaré: Con mi tarjeta de crédito ☐ American Express ☐
Visa ☐ Interbank ☐
Contra reembolso ☐ Adjunto cheque ☐
Número de mi tarjeta ____
Fecha de caducidad ____
NOMBRE ____
DIRECCION ____
CIUDAD ____
C.P. ____ PROVINCIA ____
Sin gastos de envío

BASIC DESDE CODIGO MAQUINA



A veces puede existir la necesidad de acceder al intérprete Basic desde una rutina en código máquina sin necesidad de abandonar ésta. El programa de este artículo le permitirá realizar esto de forma sencilla, tal y como podrá ver con el ejemplo que le expondremos.

El programa, paso a paso

Esta rutina en código/máquina se basa fundamentalmente en la creación de un comando Basic para que sea ejecutado, introduciéndolo previamente en el área de editar (cuyo comienzo está contenido en la variable del sistema 'E LINE').
Lo primero que realiza es limpiar lo que hay por encima del área de edición, incluido el espacio de trabajo y la pila del calculador. A continuación, reserva espacio para el comando en el área de editar y copia en ella los códigos que forman el comando, contenidos en una tabla.
El siguiente paso consiste en tratar la variable del sistema 'ERR SP' para que pueda efectuarse el retorno a la rutina.
Inmediatamente, se produce una llamada a la rutina ROM que evalúa una línea BASIC (en la dirección 7050 d.), y al volver, limpia las áreas de trabajo y comprueba si se ha producido un error, retornando al intérprete BASIC si así ha sido.
Los comentarios del listado fuente en ensamblador le informarán con más precisión de cada uno de los pasos del programa.

Particularidades de la rutina

La rutina es plenamente efectiva y útil si se llama desde otro programa en código

máquina, y una vez resuelta, aquél seguirá su curso. De todos modos, no existe ningún inconveniente en llegar a ella desde el BASIC (RANDOMIZE USR 65000), sobre todo si desea comprobar su correcto funcionamiento.

Cuando se accede a ésta, el par de registros HL apunta al comienzo de una tabla donde están introducidas las órdenes BASIC que se desea ejecutar. Por supuesto, deberá escribirlas en forma de TOKENS y códigos ASCII. Veamos un ejemplo sencillo para que lo pueda entender perfectamente. Si desea que se ejecute el comando: *PRINT "N. = "; X* deberá definirlo en la tabla mediante pseudomnemónicos DEFB con los siguientes datos: (debajo de cada uno de ellos figuran los caracteres u órdenes que representan).

245 34 78 46 61 34 59 88 13
PRINT " N . = " ; X

El valor 13 del final de la serie equivale a ENTER y debe introducirse siempre como último dato.

Hay que tener en cuenta que cuando representamos números, el Z-80 los almacena con un formato binario de 5 bytes. Si queremos evitar esto, la función VAL puede servirnos, escribiendo los números de la forma VAL «número» (ej.: PRINT VAL «128»).

Ejemplo práctico de utilización del programa

Para que observe de qué forma puede aplicar la rutina, hemos colocado en la tabla del listado en Ensamblador, los códigos pertenecientes a las siguientes órdenes BASIC:

INPUT a$: FOR k=VAL "1" TO LEN a$: POKE VAL "29999" + k, CODE a$(k): NEXT k: POKE k + VAÑ "30000", VAL "0".

Supongamos que ha llamado la rutina desde su programa en código/máquina. Al ejecutarse ésta, entra en funcionamiento el comando INPUT. Después de introducido el texto, éste se almacena a partir de la dirección 30000 (el 0 final es para indicar el fin de la cadena). Acto seguido, su programa seguirá su curso normal a partir de la dirección donde realizó la llamada.

Orlando Araujo Martín

PROGRAMA CARGADOR BASIC

Introduce el código máquina en memoria, verifica los datos y realiza una pequeña demostración de sus posibilidades

```
  10  DATA 33,49,254,205,240,253
,207,255,229,205,176,22,42,89,92
,1
  20 DATA 130,0,205,85,22,35,235
,225,1,128,0,237,176,253,203,48,
254
  30 DATA 253,54,0,255,253,54,10
,1,42,61,92,229,33,33,254,229,23
7
  40 DATA 115,61,92,205,138,27
  50 DATA 225,225,34,61,92,205,1
76,22,58,58,92,254,255,194,3,19,
201
  70 DATA 238,97,36,58,235,107,6
1,176,34,49,34,204,177,97,36,58,
244
  80 DATA 176,34,50,57,57,57,57,
34,43,107,44,175,97,36,40,107,41
,58,243,107,58
  90 DATA 244,107,43,176,34,51,4
8,48,48,48,34,44,176,34,48,34,13
 100 CLEAR 64999: LET X=0: FOR I
=65000 TO 65127: READ A: POKE I,
A: LET X=X+A: NEXT I
 110 IF X<>14340 THEN  PRINT "ER
ROR EN DATAS": STOP
 120 REM
 130 REM
 140 REM
 150 REM DEMOSTRACION
 160 PRINT "PULSE UNA TECLA PARA
 PONER EN   MARCHA EL CODIGO MAQ
UINA E INTRODUZCA UN TEXTO A CON
TINUACION": PAUSE 0
 180 RANDOMIZE USR 65000
 190 CLS : PRINT "PULSE UNA TECL
A PARA COMPROBAR  SI SU TEXTO SE
 HA ALMACENADO    BIEN A PARTIR
DE LA DIRECCION   30000"
 200 PAUSE 0
 300 FOR I=30000 TO 30000+LEN A$
: PRINT I,CHR$ (PEEK I): NEXT I
```

LISTADO FUENTE EN ENSAMBLADOR

La tabla contiene los códigos de las siguientes instrucciones BASIC: INPUT a$: FOR K=VAL "1" TO LEN a$: POKE VAL "29999" + K, CODE a$ (K): NEXT K: POKE K + VAL "30000", VAL "0"

```
  10          ORG  65000
  20          ENT  $
  25          LD   HL,TEXTO    ;AP
UNTA AL CODIGO BASIC SITUADO EN
TABLA
  30          CALL CODIGO      ;LL
AMDA A LA RUTINA
  40          RST  8          ;RETO
RNO AL MODO COMANDO
  50          DEFB 255
  60 CODIGO PUSH HL
  70          CALL 5808       ;USA
UNA RUTINA ROM PARA LIMPIAR BUFF
ER
  80          LD   HL,(23641);APU
NTA A LA VARIABLE DEL SISTEMA 'E
 LINE'
  90          LD   BC,130     ;RES
ERVA 130 BYTES EN EL BUFFER DE E
DICION
  100         CALL 5717       ;LLA
MA A LA RUTINA 'MAKE ROOM'
  110         INC  HL
  120         EX   DE,HL      ;CAR
GA HL EN DE PARA EL LDIR
  130         POP  HL         ;APU
NTA AL COMIENZO DE LA TABLA
  140         LD   BC,128     ;LON
GITUD MAXIMA DEL TEXTO
  150         LDIR ;CARGA EL TEXT
O EN EL BUFFER DE EDICION
  160         SET  7,(IY+48)
  170         LD   (IY+0),255 ;LI
MPIA ERRORES ANTERIORES
  180         LD   (IY+10),1  ;ES
TABLECE LA VARIABLE NSPPC A LA P
RIMERA SENTENCIA
  190         LD   HL,(23613) ;GU
ARDA EL VALOR DE 'ERR SP' PARA
  200         PUSH HL             ;RE
CUPERARLO MAS ADELANTE
  205         LD   HL,VOLVER
  207         PUSH HL
  210         LD   (23613),SP ;CA
RGA EL STACK POINTER EN 'ERR SP'
  220         CALL 7050           ;LL
AMADA A LA RUTINA DE EVALUACION
DE LINEAS BASIC
  230         POP  HL             ;SI
 VOLVEMOS AQUI DESECHAMOS ESTE V
ALOR
  240 VOLVER POP  HL              ;SI
 HAY UN ERROR SE EJECUTA ESTA SU
BRUTINA
  250         LD   (23613),HL
  260         CALL 5808           ;LI
MPIA LOS BUFFERS
  270         LD   A,(23610)  ;SI
 HA HABIDO UN ERROR EVALUA 'ERR
NR'
  280         CP   255            ;SI
 EL CONTENIDO ES 255 NO HAY ERRO
R
  290         JP   NZ,4867    ;SI
 NO,SALTO AL BASIC
  300         RET  ;VUELVE AL PRO
GRAMA QUE LLAMO LA RUTINA
  310 TEXTO  DEFB 238   ;INPUT
  525         DEFB 97    ;a
  720         DEFB 36    ;$
```



```
955     DEFB 58    ;:          6545    DEFB 97    ;a
1170    DEFB 235   ;FOR        6760    DEFB 36    ;$
1385    DEFB 107   ;k          6975    DEFB 40    ;(
1600    DEFB 61    ;=          7190    DEFB 107   ;k
1815    DEFB 176   ;VAL        7405    DEFB 41    ;)
2030    DEFB 34    ;"          7620    DEFB 58    ;:
2245    DEFB 49    ;1          7835    DEFB 243   ;NEXT
2460    DEFB 34    ;"          8050    DEFB 107   ;k
2675    DEFB 204   ;TO         8265    DEFB 58    ;:
2890    DEFB 177   ;LEN        8480    DEFB 244   ;POKE
3105    DEFB 97    ;a          8695    DEFB 107   ;k
3320    DEFB 36    ;$          8910    DEFB 43    ;+
3535    DEFB 58    ;:          9125    DEFB 176   ;VAL
3750    DEFB 244   ;POKE       9340    DEFB 34    ;"
3965    DEFB 176   ;VAL        9555    DEFB 51    ;3
4180    DEFB 34    ;"          9770    DEFB 48    ;0
4395    DEFB 50    ;2          9985    DEFB 48    ;0
4610    DEFB 57    ;9          10200   DEFB 48    ;0
4825    DEFB 57    ;9          10415   DEFB 48    ;0
5040    DEFB 57    ;9          10630   DEFB 34    ;"
5255    DEFB 57    ;9          10845   DEFB 44    ;,
5470    DEFB 34    ;"          11060   DEFB 176   ;VAL
5685    DEFB 43    ;+          11275   DEFB 34    ;"
5900    DEFB 107   ;k          11490   DEFB 48    ;0
6115    DEFB 44    ;,          11705   DEFB 34    ;"
6330    DEFB 175   ;CODE       11920   DEFB 13    ;ENTER
```


DISPONEMOS DE TAPAS ESPECIALES PARA SUS EJEMPLARES DE

Todospectrum

SIN NECESIDAD DE ENCUADERNACION



Para hacer su pedido, rellene este cupón HOY MISMO y envíelo a:

Todospectrum Bravo Murillo, 377
Tel. 733 96 62 - 28020 MADRID

Por favor envíenme .... tapas para la encuadernación de mis ejemplares de TODOSPECTRUM, al precio de 650 pts. más gastos de envío.

El importe lo abonaré
□ POR CHEQUE □ CONTRA REEMBOLSO □ CON MI TARJETA DE CREDITO □ AMERICAN EXPRESS □ VISA □ INTERBANK

Número de mi tarjeta:

Fecha de caducidad .... Firma

NOMBRE ....

DIRECCION ....

CIUDAD .... C. P. ....

PROVINCIA ....

(cada tapa es para 6 ejemplares)



APARTADO DE CORREOS

Dirige tus cartas a:
Todospectrum
Bravo Murillo, 377, 5.º-A
28020 Madrid

LENGUAJE MAQUINA, FUNCIONES Y AJEDREZ

Dada la falta de una sección de «Preguntas y respuestas» en el suplemento QL MAGAZINE ha creído conveniente remitir mi carta a la que trata sobre cuestiones del Spectrum. Quisiera que me respondieran a lo siguiente:

¿Existe alguna publicación sobre el lenguaje máquina del QL?

¿Cómo podría introducir desde el teclado una función cualquiera sin necesidad de cambiar una línea del programa? (Esto vale también para el Spectrum, pues tuve el mismo problema con él).

Tengo el programa de ajedrez de PSION y mi problema es que cuando al final de la carga me pide que inserte el cartucho original para iniciar el juego, resulta que no funciona. El microdrive funciona perfectamente, y puede asegurar que no he manejado incorrectamente el cartucho, sino sacar copias de seguridad y cargar desde ellas, pero lo compré hace 6 meses y temo que no conservo la garantía. ¿Merece la pena ir al establecimiento a que me lo cambien o hay alguna solución mejor?

José L. Collado
Zaragoza

Debido al poco volumen de cartas referentes a QL que llegan a la redacción, no creemos que sea necesaria la creación de una sección de este tipo en el suplemento. Sin embargo, intentaremos contestar desde aquí a todas las cartas que lleguen a TODOSPECTRUM, se refieran a uno o a otro ordenador.

Sí existe un libro que trata del lenguaje máquina del 68008 en el QL. Se trata de *Assembly Language Programmilng on the Sinclair QL* de Andrew Pennell y la editorial inglesa Sunshine Books, aunque, como habrás adivinado por el título, está escrito en inglés, y, por lo que sabemos, no hay ningún indicio de que vaya a ser traducido al castellano próximamente. Aunque menos específico puede serte muy útil *QL, Programación Avanzada* de Adrian Dickens y la editorial (esta vez española) RA-Ma, que comenta más superficialmente las instrucciones del assemble 68008 pero contiene importantes datos sobre el sistema operativo QDOS, mapa de memoria, traps, vectores, etc.

Tanto en el QL como en el Spectrum, las funciones definidas por el usuario deben estar incluidas dentro de los listados para que puedan ser ejecutadas. Tienes por tanto dos opciones, o encontrarles un hueco dentro del BASIC o prescindir de ellas.

En cuanto a tu problema con el Chess, se trata sin duda de que el cartucho original se ha deteriorado por una u otra causa. Prueba en la tienda donde lo adquiriste por si pudieran cambiártelo, quizás aún lo consigas.

U.D. G.s DISCRIMINADOS

Poseo un ZX Spectrum de 16 K, y me gustaría saber por qué no funciona la función SCREEN$ (X, Y) cuando se trata de identificar un caracter definido por el usuario, en cambio cuando se trata de hacer lo mismo con un caracter predefinido por la máquina todo funciona a la perfección. ¿Cómo podría conseguir utilizar esta función con caracteres definidos por el usuario?

Ferrán Font
Badalona

Suponemos que los señores que programaron la ROM del Spectrum sabrán los motivos que les llevaron a «dejar a medias esta función, incluida la errata de salida de la rutina de la ROM SCANNING-SCREEN$, que, al hacer un salto a STACK-STORE-$ en lugar de el correspondiente RET, produce un almacena-

miento por duplicado de la cadena resultante.

Pero, como de nada vale lamentarse, aquí tienes una rutina en código máquina (SCR$_G) que te resolverá el problema.

Pokeadas las coordenadas X e Y en ORG+1 y ORG+2, la rutina podrá ser llamada a la dirección que ocupe (en el listado 65280). El código del carácter que se encuentre en la posición señalada quedará en el par BC, por lo que puede utilizarse directamente de la forma que más nos convenga, por ejemplo: IF USR xxxxx = 144 THEN GOT nn (siendo xxxxx la dirección donde coloquemos la rutina). Si el caracter no es reconocido la rutina devolverá un cero.

Si no dispones de ensamblador ni conocimientos de este lenguaje no debes preocuparte, puedes teclear el programa cargador BASIC, que se encargará de la carga y comprobación de las líneas DATA. Caso de que, una vez ejecutado, aparezca el mensaje «ERROR en DATAs», deberás revistar los valores de estas líneas y repetir la ejecución. Una vez pasado este test, y antes de ejecutar el código máquina, deberás salvar el programa en cinta con:

SAVE "NOMBRE" CODE 65280,73

Cuando quieras volver a cargarlo deberás teclear:

CLEAR 65279: LOAD "" CODE

LISTADO 1

```
10 ;------------------------
20 ; SCREEN$ (x,y) DE UDGs
30
40 ;ENT: BC=COORDENADAS
50 ;SLD: C=CODIGO UDG O 0 SI
60 ;          NO ENCONTRADO
70 ;------------------------
80
90
100         ORG   #FF00
110
120         LD    BC,#0000
130 SCR$_G
140         LD    HL,(#5C7B)
150         LD    A,C
160         RRCA
170         RRCA
180         RRCA
190         AND   #E0
200         XOR   B
210         LD    E,A
220         LD    A,C
230         AND   #18
240         XOR   #40
250         LD    D,A
260         LD    B,#15
270 SCR$G1
280         PUSH  BC
290         PUSH  DE
300         PUSH  HL
310         LD    A,(DE)
320         XOR   (HL)
330         JR    Z,SCR$G2
340         INC   A
350         JR    NZ,SCR$G4
360         DEC   A
370 SCR$G2
380         LD    C,A
390         LD    B,#07
400 SCR$G3
410         INC   D
420         INC   HL
430         LD    A,(DE)
440         XOR   (HL)
450         XOR   C
460         JR    NZ,SCR$G4
470         DJNZ  SCR$G3
480         POP   BC
490         POP   BC
500         POP   BC
510         LD    A,#80
520         SUB   B
530         LD    BC,#0001
540         RST   #30
550         LD    (DE),A
560         JR    SCR$G5
570 SCR$G4
580         POP   HL
590         LD    DE,#0008
600         ADD   HL,DE
610         POP   DE
620         POP   BC
630         DJNZ  SCR$G1
640         LD    A,#DB
650 SCR$G5
660         ADD   A,#25
670         LD    C,A
680         RET
```

LISTADO 2

```
10 REM cargador SCR$_G

20 CLEAR 65279
30 LET c=0
40 FOR n=0 TO 72
50 READ a
60 LET c=c+a
70 POKE 65280+n,a
80 NEXT n
90 PRINT FLASH 1;" ERROR EN DA
TA " AND C<>6735
100
110 DATA 1,0,0,42,123
120 DATA 92,121,15,15
130 DATA 15,230,224,168
140 DATA 95,121,230,24,238
150 DATA 64,87,6,21,197
160 DATA 213,229,26,174,40
170 DATA 4,60,32,26,61
180 DATA 79,6,7,20,35
190 DATA 26,174,169,32,15
200 DATA 16,247,193,193,193
210 DATA 62,128,144,1,1
220 DATA 0,247,18,24,11
230 DATA 225,17,8,0,25
240 DATA 209,193,16,211,62
250 DATA 219,198,37,79,201
```

SIN ANIMO DE CRITICAR

Muy Sr. mío:

Soy asiduo lector de la revista TODOSPECTRUM, y habiendo visto en el número 16, correspondiente a diciembre, el anuncio de comercialización del programa Starmouse (dibujando con ratón), y dado el interés que para mí tenía, a través de la firma El Corte Inglés de Oviedo, lo pedí. Tardó muy pocos días en llegar a esta ciudad. Una vez cargado en el QL, resultó ser un programa de gran utilidad.

El gran fallo, que es imperdonable, y que fue motivo de la devolución del importe de mi compra, después de infinitas gestiones con la firma Puricorp de Guadalajara, con Investrónica, y con los autores del programa, es que al parecer sólo se puede imprimir lo presentado en la pantalla con un único tipo de impresora, cuya marca no tengo ningún interés de conocer. Mi impresora es una ADMATE DP-100.

Si le parece oportuno puede transmitir el contenido de este escrito a los lectores de sus prestigiosa revista, para que nos les pase lo que a mí. Y menos mal, que en mi caso, la firma comercial El Corte Inglés, dada su solvencia, no puso el menor inconveniente en la devolución del importe, y fue ella la que investigó en las firmas antes mencionadas sobre la incongruencia del caso.

El problema concreto que aparece con las impresoras «non gratas» es que se saltan un renglón y medio, que queda sin entintar. Como muestra le adjunto un recorte de lo que en pantalla eran círculos rellenos de color o trama. Con el resto de líneas, rayados, etc., trazados con el ratón, ocurre exactamente lo mismo, salen en franjas con intervalos de renglón y medio en blanco.

Sin otro particular, y rogándole sepa disculpar el haberle sustraído parte de su tiempo, quedo suyo affmo.

V. Enríquez Fernández
Oviedo

EL CORCHO

Vendo interface impresora centronics/RS232 Ventamatic, por cambio de ordenador. Regalo cable para conectar a impresora, valorado en 3.000 pesetas, y programa Tasword Two adaptado al interface. Todo por 7.000 pesetas. Preferentemente en Barcelona. Llamar al (93) 422 04 66 entre las 22,00 y 23,00.

Vendo Spectrum 48K, como nuevo, con todos sus accesorios, por 19.000 pesetas. También vendería interface programable de sonido y joystick. Teléfono: (968) 26 38 39. Murcia.

Vendo calculadora Casio FX-3600P, programable, científica, análisis regresión, etcétera. Además regalo juegos actuales. Todo por 7.000 pesetas negociables. También video juegos Atari por 10.000 pesetas con ocho cartuchos de regalo. Alejandro Esteve. Avda. Burjasot, 112 9. 46009 Valencia. Teléfono: (96) 349 76 04. Horas comida.

Estoy interesado en conseguir los programas «Doomdark's Revenge» y «D-Day». Escribir a Ignacio López de Torre. Calle del Río, 27. Miranda de Ebro (Burgos).

Se venden programas para Spectrum, por cambio de ordenador. Escribid a Juan Carlos Hernández. Calle Condado de Treviño, 75, 6-dcha. Miranda de Ebro (Burgos).

Club Game and Basic: Información, trucos, pokes, BASIC, código máquina... gratuito. Paco Capeáns Outeiro. Calle Carmen, 10. Negreira (La Coruña).







¡ATENCION! Con el ejemplar de MARZO 86

PC MAGAZINE le regala un DISKETTE DEMO DE PC-TEXT-PC-BASE.
(Obsequio de Quinta Generación)

Además en este número:

GRAFICOS:
Hardware y Software para hacer que el ordenador personal transforme en imágenes sus ideas y cifras, etc.

Compatible

SPERRY PC/HT

Software
ENCORE! HOJA FINANCIERA

Programación
VTREE, UN ARBOL DE VERDAD



ESTOS TEMAS Y MUCHOS MAS EN EL NUMERO DE MARZO

¡COMPRELA HOY MISMO!

ASI TRABAJA EL QL HECHO PARA NOSOTROS







QL QUILL. TRATAMIENTO DE TEXTOS.

QL Quill muestra en pantalla exactamente cómo aparecerá su escrito cuando esté impreso. Olvídese de pesadas memorizaciones de comandos. Con QL Quill tiene siempre todo lo necesario en pantalla. Escriba en negrita, corte, una, fije tabuladores, márgenes, sobreescriba... después de todo esto su QL le dirá, además, cuántas palabras ha escrito.

QL ARCHIVE. BASE DE DATOS

Organizar su agenda o poner al día su fichero se convierte en un trabajo agradable. QL Archive resuelve rápidamente su problema con un sistema de archivo de gran facilidad de uso, usando un lenguaje aún más sencillo que el BASIC.

Con un mínimo de práctica puede crear desde un simple directorio a una extensa base de datos. Podrá determinar relaciones, extraer datos, reorganizarlos, hacer cálculos estadísticos y contables.

Con QL Archive podrá crear un formato personalizado para sus informes.







QL ABACUS. HOJA ELECTRONICA DE CALCULO

Este programa ha superado con mucho las tradicionales hojas de cálculo de otros ordenadores. Con QL Abacus escriba simplemente "costes de personal", o "personal", o "pers" para encontrar esa celdilla. QL Abacus también le ayuda a decidir lo que hay que hacer mediante diferentes sugerencias, dentro de una amplia gama de posibilidades. Podrá, además, crear hojas electrónicas de cálculo con más de 6.000 celdas individuales.

Podrá mostrar ventanas múltiples, variar la distancia entre columnas...

QL EASEL. GRAFICOS

Olvídese de construir tablas de valores o enfrentarse con situaciones dudosas antes de ver algún resultado. Con QL EASEL podrá desde el principio crear gráficos. Inmediatamente creará diagramas de barras, filas, líneas rellenas, sectores, barras sobrepuestas...

Todo ello simplemente pulsando una tecla.

Introduzca textos donde quiera. Dentro o cerca de su diagrama. Más aún, mueva el texto por la pantalla para ver dónde queda mejor... en el color que prefiera.



investronica

Tomás Bretón, 60. Telf. (91) 467 82 10. Telex 23399 IYCO E. 28045 Madrid
Camp, 80. Telf. (93) 211 26 58-211 27 54. 08022 Barcelona